all rights reversed

HACKER

Αύγουστος 2012 # Τιμή τεύχους 6,40€

τεύχος 011 - Lazy August Edition

Τα BOTNETS ξυπνάνε! [*]



Στήστε τον δικό σας IRC server (με τα όλα του)

Μάθετε για τα botnets, μετά φτιάξτε και το δικό σας!

Μεταμορφώστε απλούς routers σε τιτάνες της δικτύωσης

Αναβαθμίστε τη φωτογραφική σας μηχανή (like, δωρεάν)

[*] Νομίζατε ότι θα 'ναι ένα αδιάφορο, καλοκαιρινό τεύχος, ε;

tefhos 11o

Περιεχόμενα

και οδηγίες χρήσης

60 πανίσχυρος και ικανότατος router, για το οικιακό μας δίκτυο

...γιατί το router-part του modem/router που πήραμε από τον ISP, είναι πραγματικά για γέλια

68 δημιουργία ενός botnet, από την αρχή!

θέλετε να δείτε πώς είναι να κάθεστε
στη θέση του botmaster;
προσέξτε, μπορεί να σας αρέσει επικίνδυνα πολύ!

84 βάλτε μυαλό στον ανεμιστήρα, παρτ του

για να γίνει το ανεμιστηράκι έξυπνο
χρειάζεται software,
όπως εξάλλου και ο υπολογιστής που το ελέγχει

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) εδώ και τώρα, πρόταση νόμου για τη μέγιστη επιτρεπτή θερμοκρασία κατά τη διάρκεια του καλοκαιριού!



Πλήρη άρθρα που συμπληρώνουν το περιοδικό. Τα εβδομαδιαία μας video casts [έρχονται, έρχονται]. Ολόκληρα blasts from the past. Τα βίντεο από τα live streams. Τα πάντα ενδιαφέροντα forums. Το σημείο συνάντησης για τους φίλους του περιοδικού. Η πιο ζωντανή κοινότητα και οι πιο ανήσυχοι, δημιουργικοί χρήστες!

deltahacker.gr

What have you hacked for me lately?

Ανησυχίες του Αυγούστου

Δεν θυμάμαι να το έχω αναφέρει ξανά, γι’ αυτό θα το αναφέρω κι εδώ: Η προετοιμασία του νέου τεύχους είναι μεγάλος αγώνας. Μεγάλος, και σχεδόν πάντα αγωνιώδης. Είναι πολλές οι λεπτομέρειες που πρέπει να κανονιστούν, απειραριθμήσιμα τα κομμάτια του παζλ που πρέπει να μπουν στη σωστή θέση. Όταν μάλιστα είναι και κατακαλόκαιρο, ο αγώνας δυσκολεύει απότομα κι απρόσμενα...

Συνεννοείσαι με τους ανά την επικράτεια συνεργάτες -- και μ’ έναν που βρίσκεται στα εξωτερικά. Κανονίζετε και συμφωνείτε στα θέματα. Διευκρινήσεις πάνε κι έρχονται. Μετά από μια περίπου βδομάδα, αρχίζουν να ‘ρχονται και τα πρώτα κείμενα. Ο αρχισυντάκτης πιάνει δουλειά, ενώ η διεύθυνση απασχολείται με άλλες εργασίες.

Μην ανησυχείτε, με λεπτομέρειες δεν θα σας κουράσω. Το νου σας: Είναι πολλές αυτές οι «άλλες εργασίες» και σίγουρα θα μπορούσα να γράψω πολλά γι’ αυτές. Από την άλλη όμως είναι κι Αύγουστος μήνας και κάνει ζέστη και πραγματικά δεν έχει πολύ νόημα να κουράζουμε τα πράγματα...

Κάποια στιγμή, αγαπητές κι αγαπητοί, κι ενώ οι μέρες του μήνα τρέχουν ανελέητα, ο αρχισυντάκτης στέλνει τα πρώτα κείμενα στη διεύθυνση. Εκείνη αρχίζει το διάβασμα, βάζει και καμιά πινελιά εδώ κι εκεί. Για κάθε κείμενο που περνάει από τη διεύθυνση, πριν πάει στο ατελιέ και στον DTP guy, περνάει πρώτα από το τμήμα της γραμματικής επιμελείας. Εκεί, η διπλοθεσίτης Υπεύθυνη Διορθώσεως και Σύμβουλος Εκδόσεως, κάνει γενναίες και φιλότιμες προσπάθειες ώστε τα κείμενα να είναι γραμμένα στα ελληνικά.

Ψέμματα: Κάνει γενναίες και φιλότιμες προσπάθειες ώστε τα κείμενα να είναι γραμμένα σε *σωστά* ελληνικά.

Σ’ αυτόν τον αγώνα, μόνη δεν είναι. Ο κος Εκδότης, ο οποίος ποτέ δεν ήταν φανατικός της γραμματικής, για έναν παράξενο λόγο πάντα είχε --και έχει-- μια μανία με τη γραμματική. Μη ρωτάτε γιατί και πώς. Απλά έχει -- κι ας μην ήταν ποτέ φανατικός της γραμματικής. Στις παρατηρήσεις της υπευθύνου διορθώσεως, λοιπόν, δίνει πάντα μεγάλη προσοχή.

Όταν είναι Αύγουστος, όμως, η μεγάλη προσοχή φέρνει στην επιφάνεια και μεγάλους προβληματισμούς -- και συνήθως με αφορμές που ποτέ δεν θα φανταζόσουν. Ακολουθεί παράδειγμα.

Ποιο είναι, σας ερωτώ, το σωστό γένος για τον/το router;

Σοβαρά ρωτάω. Μπορείτε να μου πείτε ποιο είναι το σωστό γένος; Αν σκεφτούμε «δρομολογητής», τότε σίγουρα δεν υπάρχει πρόβλημα όταν γράφεις «ο router». Αν σκεφτούμε μια συσκευή, τότε πάλι δεν υπάρχει πρόβλημα να γράφεις “το router”, “το routerάκι” κ.ο.κ.

Τελικά, όμως, τι από τα δύο ισχύει; Γιατί, με συγχωρείτε δηλαδή, αλλά και τα δύο να είναι σωστά αποκλείεται. Δηλαδή δεν αποκλείεται, αλλά η μαθηματική παιδεία που έχει η διεύθυνση δεν της επιτρέπει να θεωρεί *και* το ένα σωστό *και* το άλλο.

Δεν ξέρω. Φταίει ίσως το γεγονός ότι η διεύθυνση ξεχνά ότι ο πραγματικός κόσμος κρατάει μια κάποια «απόσταση» από τα μαθηματικά.

Όπως και να ‛χει, μετά από έντονο προβληματισμό και μπόλικη ανησυχία, η κυρία υπεύθυνη γραμματικής επιμελείας κι ο κύριος εκδότης αποφάσισαν ότι και τα δύο είναι σωστά. Το τι γράφεται ανά πάσα στιγμή, συμφώνησαν, εξαρτάται από τα συμφραζόμενα. Έτσι, στο κείμενο του καλού συνεργάτη και φίλου giannoug που θα βρείτε στο παρόν τεύχος και αφορά στο dd-wrt, αλλού γράφουμε για “τον router» κι αλλού για “το router».

Τώρα που το σκέφτομαι, για αρκετές από τις αυγουστιάτικες ανησυχίες του γράφοντα ευθύνεται ο ίδιος καλός συνεργάτης -- και φίλος, φυσικά. Να, τις προάλλες είχε ετοιμάσει, ο γράφοντας, μια λίστα από τεχνικές εργασίες που πρέπει να κάνει. Είναι από εκείνες τις εργασίες που λες στον εαυτό σου ότι είναι για τη δουλειά, στην πραγματικότητα όμως ξέρεις ότι πρόκειται για καθαρή, αγνή διασκέδαση (pure geek fun, baby!)

Παράδειγμα: Με βάση το άρθρο που έχει γράψει ο giannoug για το duply στο τεύχος 009, πρέπει τώρα να κανονίσεις ώστε να λαμβάνονται αυτομάτως ασφαλή backups του deltahacker.gr σ’ ένα άλλο VPS.

Άλλο παράδειγμα: Μιας κι ο sonic σου έχει πάρει τα μυαλά με το FreeBSD κι αφού έχεις που έχεις στήσει έναν FreeBSD fileserver με ZFS, δες πώς στο καλό μπορείς να ρυθμίσεις το σύστημα ώστε η πρόσβαση στο ZFS volume να μην είναι *τόσο* αργή!

Θέλετε κι άλλο παράδειγμα; Σίγουρα θέλετε. Έγραψε που λέτε ο giannoug (ναι, πάλι αυτός) στο τεύχος 010, για το πώς στήνεις το δικό σου (ή **τον** δικό σου) exit node, εντός του δικτύου ανωνυμίας Tor. Τώρα, έχεις που έχεις ένα VPS να κάθεται κάπου στην Καλιφόρνια, γιατί να μη στήσεις κι εσύ ένα Tor exit node, σκέφτεσαι. Οπότε κάθεσαι και το στήνεις. Και μετά από λίγο είναι έτοιμο και δουλεύει μια χαρά. Κι εκεί που δουλεύει μια χαρά, σκέφτεσαι ότι μπορείς να σνιφάρεις το διερχόμενο traffic. Όχι για κακό, αλλά για εκπαιδευτικούς και δημοσιογραφικούς λόγους -- για τη δουλειά, δηλαδή :D Αν μη τι άλλο, θέλεις να διαπιστώσεις κατά πόσο οι χρήστες του Tor ευησυχάζουν, ακριβώς επειδή βγαίνουν στο Internet μέσω του Tor.

Αυτά κι άλλα πολλά, φίλες και φίλοι, σε προβληματίζουν, σε ανησυχούν και σίγουρα σε κρατάνε απασχολημένο Αυγουστιάτικα. Γιατί ας μη γελιόμαστε: Δεν υπάρχει καλύτερη καλοκαιρινή απασχόληση για ένα geek, σωστά;

Εντάξει, υπάρχουν κι εκείνες οι διακοπές, το άδειασμα του μυαλού και η φόρτιση των μπαταριών. Κάτι μου λέει μάλιστα ότι οι περισσότεροι από εσάς θα διαβάζετε το παρόν από μια παραλία.

Ελπίζουμε να βρείτε ενδιαφέρον το αυγουστιάτικο τεύχος, όπου κι αν είστε, όπως κι αν περνάτε. Οι προβληματισμοί μας, όπως θα διαπιστώσετε, δεν του επέτρεψαν να ‛ναι άλλη μια νωχελική --και κατά βάση βαρετή-- καλοκαιρινή έκδοση.

Αντίθετα, τολμώ να πω ότι είναι ένα από τα πιο ανήσυχα τεύχη που έχουμε βγάλει.

Με την καλή έννοια.

subZraw

ουέλ μέλ

POST CAR
THE ADDRESS TO BE WRITTEN ON THIS SID





Αγαπητοί μου,

Σας γράφω από Απροσδιόριστη Τοποθεσία (ΤΜ) στο ΒορειοΔυτικό Λονδίνο. Δεν έχω ιδέα πού βρίσκομαι. Ξέρω μόνο ότι βρέχει, και ότι έχει μια στάση λεωφορείου απέναντι. Ήρθα χτες βράδυ, έχοντας πάρει, με το που κατέβηκα από το αεροπλάνο, ένα τρένο, ένα λεωφορείο, ένα μετρό, ένα δεύτερο μετρό, και ένα ακόμα λεωφορείο, το οποίο τελικά με άφησε στην στάση απέναντι.

Η διαδρομή ήταν τόσο πολύπλοκη, που μετά από λίγο έχασα τον λογαριασμό και έπαψα να την παρακολουθώ. Ήξερα μόνο την γενική κατεύθυνση.

Τώρα βρίσκομαι καθισμένος σε ένα σκαμπό, μπροστά από ένα τραπεζάκι για φορητό υπολογιστή. Πάνω στο τραπεζάκι είναι η γραφομηχανή, χωρίς την οποία, φυσικά, δεν πηγαίνω πλέον πουθενά.

Ταλαιπωρήθηκα βέβαια λίγο να τη φέρω, καθώς ούτε για χειραποσκευή μπορούσε να περάσει, ούτε με λάπτοπ έμοιαζε, ούτε χωρούσε στην κανονική μου βαλίτσα. Δεν το μετανοιώνω όμως.

Έχουμε ανέβει στη Γηραιά Αλβιώνα οικογενειακώς, ο καθένας για τους δικούς του λόγους -- ένας εκ των οποίων είναι να επισκεφθούμε τον ανιψιό.

Τον κακομοίρη τον ανιψιό, δεν ξέρω πώς τα κατάφερε έτσι, και φέτος δεν μπορούσε να αφήσει τις δουλειές του με τίποτα. Είχε ξεμείνει στο Λονδίνο να μουλιάζει από τη βροχή που, όπως μας λέει, άρχισε τον Μάιο και έκτοτε δεν λέει να σταματήσει.

Το διαπιστώνω κι εγώ. Τώρα βρέχει και η θερμοκρασία περιβάλλοντος, Ιούλιο μήνα, είναι 14 βαθμοί.

Την ιδέα για Λονδίνο την έριξε η Μαιρούλα. «Θα πάω να δω τους Ολυμπιακούς», μας ανακοίνωσε περιχαρής τις προάλλες, ενώ ήμασταν καθισμένοι στο τραπέζι και περιμέναμε να σερβιριστούμε παϊδάκια τζαλφρέζι με ρύζι μπασμάτι, φύλλα νάαν και κουρκουμά.

«Τι έγινε;», την κορόιδεψα (καλόκαρδα) εγώ, «Μας τελείωσε ο Όλυμπος, και πιάσαμε τους Ολυμπιακούς;»

«Μην είσαι είρων», ακούστηκε από την κουζίνα η μητέρα της εξαδέλφης, που είναι Ινδουίστρια. «Για πολλούς οι Ολυμπιακοί είναι η σημαντικότερη διεθνής αθλητική διοργάνωση.»

«Κοιτάχτε...», της απάντησα εγώ λίγο μουδιασμένα γιατί πείναγα και δεν ήθελα να διακινδυνεύσω τα παϊδάκια τζαλφρέζι με το ρύζι μπασμάτι, τα φύλλα νάαν και τον κουρκουμά. «Ως διεθνής αθλητική διοργάνωση οι Ολυμπιακοί είναι πράγματι η σημαντικότερη. Αλλά ως εκεί. Πόσο σημαντική δηλαδή μπορεί να είναι μια αθλητική διοργάνωση;»

Από την κουζίνα ακουγόταν μόνο ο ήχος της κουτάλας στα πιάτα.

«Είναι σαν να μου λέτε ότι το 'Έγκλημα και Τιμωρία' είναι η δημοφιλέστερη τουρκική σαπουνόπερα σήμερα», συνέχισα. «You state the facts.

Έκαστος στο είδος του, που λέει κι ο Λουμίδης.»

«Τι 'Έγκλημα και Τιμωρία' καλέ; 'Έρωτας και Τιμωρία' είναι!», κορόιδεψε (καλόκαρδα) η Μαιρούλα και γέλασε. Γελάσαμε όλοι.

Ήμασταν, όπως βλέπετε, μια χαρούμενη παρέα.

Τρώγοντας τα παϊδάκια τζαλφρέζι το αποφασίσαμε σχετικά εύκολα. Εγώ θυμήθηκα ότι και πριν από οχτώ χρόνια πάλι στο Λονδίνο ήμουν, πάλι επί Ολυμπιακών -- τότε για να τους αποφύγω· η μητέρα της εξαδέλφης θυμήθηκε ότι είχε ακούσει πως έχει κάτι μαγαζιά με παραδοσιακά σάρι στην Μπρικ Λέιν, και αποφάσισε ότι πολύ θα ήθελε να τα δει· η κυρία Καίτη θα ήθελε, επιτέλους, να δει τι είναι αυτό το Λονδίνο που λένε όλοι· και η εξαδέλφη θα πήγαινε οπουδήποτε για να γλυτώσει από τον καύσωνα, γιατί όχι και στο Λονδίνο; Ένας παραπάνω λόγος, σκέφτηκα εγώ.

Βέβαια τώρα που κάθομαι ντυμένος σαν το κρεμμύδι, με κάλτσες, μακρύ παντελόνι, μπλουζάκι, πουκάμισο και κολλεγιακό, και παρόλα αυτά τουρτουρίζω, το ξανασκέφτομαι. Αλλά πλέον είναι αργά.

Στο σπίτι επικρατεί ηρεμία. Ακούγεται μόνο η βροχή που χτυπάει τα παράθυρα, και ο άνεμος που σείει τα δέντρα και τους θάμνους στον πίσω κήπο.

Summer in the city.

Φυσικά δεν σκοπεύω να πάω στους Ολυμπιακούς. Τους είχα αποφύγει προ οκταετίας, και είμαι αποφασισμένος να τους αποφύγω και φέτος.

Το Λονδίνο βοηθάει. Δεν ξέρω αν έχετε πάει, αλλά είναι περίεργη πόλη. Εκτός από το Κέντρο, που θυμίζει ένα κανονικό (αν και τεράστιο) κέντρο ευρωπαϊκής πόλης με ποτάμι, το Λονδίνο μοιάζει με τεράστιο απλωμένο χωριό. Παντού χαμηλά σπιτάκια, παντού μικρές κεντρικές πλατείες και/ή μικροί εμπορικοί δρόμοι, και παντού μικρά μαγαζιά (εκτός από τα πολυκαταστήματα και τα σουπερμάρκετ) που πουλάνε φρούτα, λαχανικά, μπαχαρικά, ρούχα, κινητά τηλέφωνα, τηλεκάρτες, εφημερίδες, και κεμπάμπ. Υπάρχουν επίσης καφετέριες, χαμπουργκεράδικα, κουρεία, παμπ (που στην Αγγλία παίζουν το ρόλο των καφενείων), παλαιοπωλεία και στοιχηματάδικα.

Και σπιτάκια. Παντού τα ίδια σπιτάκια. Με κήπους.

Το έμμεσο πλεονέκτημα όλης αυτής της άπλας είναι ότι στο Λονδίνο η



ζωή συνεχίζεται ό,τι και να συμβεί. Μας έλεγε ο ανιψιός ότι η μόνη φορά που σταμάτησαν τα πάντα ήταν την ημέρα των βομβιστικών επιθέσεων τον Ιούλιο του 2005 -- και μόνο κατά την μέρα εκείνη. Αμέσως μετά τα πράγματα επέστρεψαν στο κανονικό.

Η μόνη άλλη μέρα που το Λονδίνο παραδοσιακώς παραλύει, είναι τα Χριστούγεννα. Δεν υπάρχει καμία απολύτως συγκοινωνία, μόνο ταξί, και όλα τα μαγαζιά είναι κλειστά. Άντε να βρεις κανένα μανάβικο και κανένα ψιλικατζήδικο στις γειτονιές.

Και βέβαια δεν κυκλοφορεί κανείς. Όλοι είναι σε κάποιο σπίτι και περνούν τις Άγιες μέρες οικογενειακά. Πιο σωστά: Όλοι *οφείλουν* να είναι σε κάποιο σπίτι. Αν δεν έχεις πού να πάς, καλύτερα να μην το πεις παραέξω γιατί οι άλλοι θα σκεφτούν ότι κάτι πάει στραβά με σένα και θα αρχίσουν να σε αποφεύγουν.

Πώς το λένε το φαινόμενο; Αυτοεκπληρούμενη προφητεία.

Διότι αν δεν σε έχουν καλέσει πουθενά, και εσύ δεν το πεις, τότε δεν θα σε καλέσουν ούτε αυτοί που μπορεί και να σε καλούσαν, με αποτέλεσμα και αυτοί να συνηθίσουν στην απουσία σου, και εσύ να συνηθίσεις στην μιζέρια σου. Και άντε μετά να ξεκολλήσεις.

Είχα δει μια ταινία, πέρσι, με θέμα ακριβώς αυτό. «Dreams of a Life». Ντοκυμαντέρ ήτανε και μου το είχε συστήσει ο ανιψιός. Φυσικά δεν παίχτηκε στην Ελλάδα· την βρήκα στα τόρεντ.

Ήταν για κάποια νεαρή κοπέλα που έμενε μόνη της, κάπου στο Βόρειο Λονδίνο. Πέθανε από παθολογικά αίτια ενώ πακετάριζε δωράκια κάποια χρονιά παραμονές Χριστουγέννων. Έμεινε εκεί, νεκρή, με το ψαλίδι στο χέρι, μπροστά στην αναμμένη τηλεόραση, για τρία χρόνια.

Για τρία ολόκληρα χρόνια κανείς δεν την αναζήτησε. Για τρία ολόκληρα χρόνια η τηλεόραση συνέχιζε να παίζει χωρίς κανείς να απορήσει.

Όταν άνοιξαν το σπίτι, βρήκανε έναν σκελετό, ένα ψαλίδι, χαρτί περιτυλίγματος, και τα Χριστουγεννιάτικα δωράκια που είχε προλάβει να ετοιμάσει.

Το ντοκυμαντέρ προσπαθούσε να ανιχνεύσει πώς έγινε αυτό, πώς έφτασαν εκεί τα πράγματα.

Τώρα θα μου πείτε, τι κάθομαι και σας γράφω καλοκαιριάτικα. Από μια άποψη έχετε δίκιο. Αλλά κι αυτός ο καιρός, δεν βοηθάει βλέπετε.

Αυτό που κυρίως θέλω να πω είναι κάτι πολύ απλό, αδυσώπητο, και απίστευτα τρομακτικό όταν το συνειδητοποιείς. Είναι το εξής:

Κάποια πράγματα πάνε στραβά, και κάποια όχι. Αλλά αν είναι να πάνε στραβά, δεν υπάρχει κανένα όριο στο πόσο στραβά μπορούν να πάνε. Δεν υπάρχει καμία ασφαλιστική δικλείδα. Ούτε η κοινή λογική, ούτε η ωριμότητα, ούτε η γνώση, ούτε η πληροφόρηση, ούτε η τεχνολογία, ούτε η συμπόνια των άλλων, ούτε τίποτα. Αν τα αφήσεις ανεξέλεγκτα, τα πράγματα μπορεί να πάνε πολύ στραβά.

Έχει συμβεί και ξανασυμβεί. Και θα ξανασυμβεί. Αναπόφευκτα. Το θέμα είναι, όσο περνάει από το χέρι μας, να μην ξανασυμβεί σε μας.

Να το θυμάστε αυτό, αγαπητοί μου, όταν σας ξαναπεί κάποιος ότι εντάξει, ρε παιδί μου, δεν τρέχει και τίποτα που έπιασαν εφτά τοις εκατό οι αβγοκέφαλοι τραμπούκοι, θλιβερό είναι αλλά εντάξει, ή ότι η Ευρώπη δεν πρόκειται με τίποτα να αφήσει την Ελλάδα να καταρρεύσει.

Να το θυμάστε.

Σας ασπάζομαι,
Θείος Ακάκιος

Skill: Beginner

Tags: bots, botnets, zombies, command and control, P2P, peer-to-peer, IRC, Internet Relay Chat, DNS, single-flux, double-flux, fast-flux

Botnets:

Φοβεροί, τρομεροί αλλά και συναρπαστικοί στρατοί από zombies!

Πρόκειται για το όνειρο των απανταχού κακόβουλων χάκερ (blackhats) και τον εφιάλτη των μεγαλύτερων εταιρειών ασφαλείας. Μικρές ομάδες έχουν τη δυνατότητα να δίνουν εντολές σε εκατοντάδες χιλιάδες υπολογιστές η καθεμιά, οι οποίοι υπακούν αδιαμαρτύρητα! Αλλά ο έλεγχος αυτός δεν είναι για καλό σκοπό. Αναφερόμαστε φυσικά στα botnets. Αλήθεια, πώς θα σας φαινόταν αν είχατε υπό τον έλεγχό σας δεκάδες ή εκατοντάδες υπολογιστές, οι οποίοι πρόθυμα θα εκτελούσαν τις εντολές σας;

by Ion

Ακούγεται γοητευτικό και το ξέρουμε, αλλά πρόκειται για μια εντελώς παράνομη δραστηριότητα. Εμείς πάντως λέμε να εξερευνήσουμε τον κόσμο των botnets -- και μάλιστα να φτιάξουμε το δικό μας! Φυσικά, τα πάντα θα γίνουν για εκπαιδευτικούς σκοπούς και χωρίς να παραβιάσουμε κανένα (γραπτό ή άγραφο) νόμο.

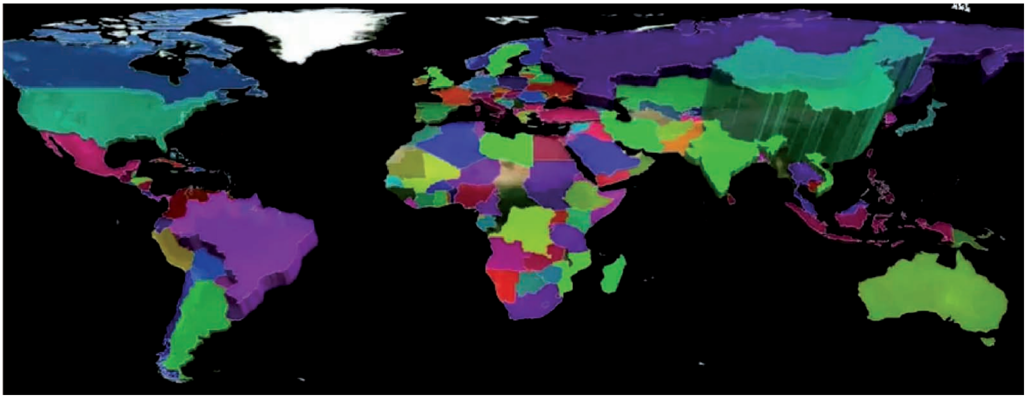
Τι είναι τα bots

Πριν περάσουμε στα ίδια τα botnets θα πρέπει να δούμε τι ακριβώς είναι ένα bot. Πρόκειται για κάποιο πρόγραμμα που τρέχει σε έναν υπολογιστή με σκοπό να εκτελεί μια ορισμένη διεργασία. Το bot είναι σε θέση να δέχεται συγκεκριμένες εντολές και να εκτελεί αναλόγως. Οι παλαιότεροι θα θυμόσαστε τα bots από το IRC (Internet Relay Chat), τα οποία ήταν απλά προγραμματάκια που βοηθούσαν στη διαχείριση ενός καναλιού συνομιλίας. Υπάρχουν bots με καλές προθέσεις, με κλασικό παράδειγμα το Google bot που επισκέπτεται αδιάκοπα εκατομμύρια σελίδες καθημερινά για να κάνει indexing του νέου περιεχομένου. Υπάρχουν όμως και κακόβουλα bots, τα οποία εκτελούν παράνομες ενέργειες όπως αποστολή SPAM mail ή επιθέσεις σε άλλα συστήματα.

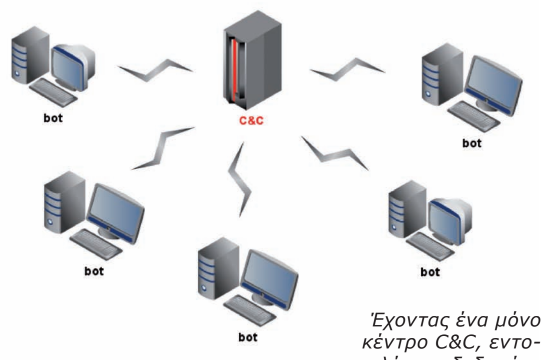
Στην πλειονότητα των περιπτώσεων, ένας υπολογιστής "γίνεται bot" χωρίς να το επιθυμεί ο κάτοχος του. Αυτό μπορεί να συμβεί με διάφορους τρόπους. Θυμάστε τα malware honeypots, που είχαμε στήσει στο deltaHacker 007; Όπως θα είχατε διαπιστώσει, υπάρχουν malware που επιτίθενται αυτόματα και χωρίς σταματημό, ενάντια σε διάφορα άλλα συστήματα. Μόλις τα καταλάβουν, εκμεταλλευόμενα μια συγκεκριμένη αδυναμία, αυτά τα συστήματα γίνονται ωραιότατα bots. Ε, ένας τρόπος "μόλυνσης" είναι αυτός.

Άλλος τρόπος είναι μέσω κοινωνικής μηχανικής, όπου κάποιος θα μπορούσε να πείσει διάφορα θύματα να εκτελέσουν ένα αρχείο στον υπολογιστή τους. Φυσικά, αυτό το αρχείο θα ήταν κάποιο trojan horse ή κάτι σχετικό, με αποτέλεσμα την ίδια κατάληξη. Αυτά θα τα δούμε παρακάτω. Το σημαντικό πάντως σε όλες αυτές τις περιπτώσεις είναι ότι οι απλοί χρήστες-θύματα ούτε συμφωνούν να γίνουν οι υπολογιστές τους bots, ούτε καν το αντιλαμβάνονται (τουλάχιστον τις περισσότερες φορές). Από εκεί κι έπειτα, κάθε bot συμπεριφέρεται ανάλογα με τις εντολές που έχει ενσωματωμένες στον κώδικά του από τον προγραμματιστή του -- και ο υπολογιστής φυσικά εκτελεί αναλόγως. Λόγω αυτής της άβουλης κατάστασης στην οποία περιέρχονται οι μολυσμένοι υπολογιστές, πολλές φορές αποκαλούνται και zombies.

Η παγκόσμια κατανομή των bots, μ' άλλα λόγια των εκατομμυρίων υπολογιστών zombies. Όσο πιο υπερυψωμένη είναι μία χώρα, τόσα περισσότερα μολυσμένα PCs βρίσκονται εκεί. Οι φίλοι μας οι Κινέζοι φαίνεται να έχουν αρκετά μεγάλο πρόβλημα. Ακολουθούν Ρωσία και Βραζιλία...



Τι είναι ένα botnet;



Έχοντας ένα μόνο κέντρο C&C, εντολές και δεδομένα μεταφέρονται γρήγορα εντός του botnet. Αν όμως το κέντρο πέσει, τότε όλα τα bots μένουν ακυβέρνητα!

Ο παραδοσιακός ορισμός ενός botnet είναι ένα δίκτυο από υπολογιστές που έχουν καταληφθεί από κακόβουλα bots. Αυτοί οι υπολογιστές ανήκουν σε ανυποψίαστους χρήστες, οι οποίοι χωρίς να το γνωρίζουν γίνονται μέλη σε αυτό το μεγάλο δίκτυο από zombies, τα οποία είναι συνδεδεμένα στο Διαδίκτυο και επικοινωνούν μεταξύ τους. Ακριβέστερα, αντί κάθε bot να εκτελεί ενέργειες “τοπικά”, εδώ μιλάμε για ένα ολόκληρο δίκτυο υπολογιστών που δέχεται εντολές απομακρυσμένα και για μαζική, συντονισμένη δράση. Προφανώς, η σύγκριση με μια διαδικτυακή στρατιά είναι αναπόφευκτη...

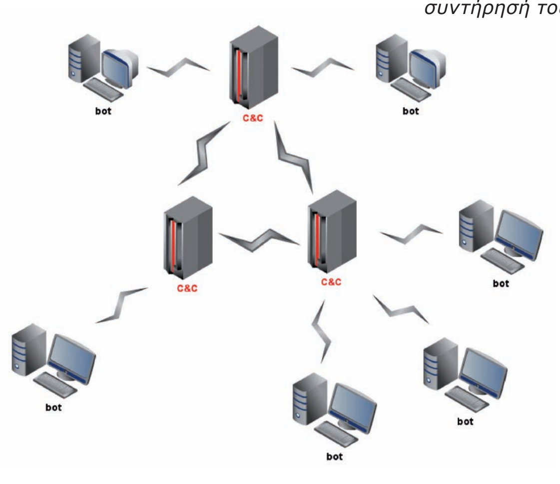
Αυτός που δίνει τις εντολές κι ελέγχει όλο το botnet είναι ο botmaster ή botherder (bot-ο-βοσκός, ένα πράγμα). Συνήθως βέβαια πρόκειται για κάποιον κακόβουλο χρήστη ή ακόμη και για κάποια ομάδα οργανωμένου εγκλήματος, η οποία και αποκομίζει χρηματικό κέρδος από τη δράση του botnet ή των botnets που ελέγχει. Ο έλεγχος αυτός γίνεται μέσω συγκεκριμένων καναλιών επικοινωνίας και χρησιμοποιώντας άλλους υπολογιστές που ονομάζονται κέντρα C&C ή αλλιώς Command & Control centers. (Σε ποιον δεν έρχεται στο μυαλό το θρυλικό Command & Conquer;)

Φυσικά, αυτό είναι ιδιαίτερα βολικό: Ο botmaster δεν δίνει εντολές απευθείας, συνεπώς η ταυτοποίησή του δεν είναι ό,τι πιο εύκολο ή άμεσο. Σκεφτείτε, π.χ., ότι είναι δυνατόν να συνδέεται στο C&C μέσω κάποιου proxy -- ή ακόμη και μέσω του δικτύου ανωνυμίας Tor. Οι εντολές που φτάνουν στο C&C εκπέμπονται στα zombies κι ενδέχεται να αφορούν σε μια επίθεση DDoS, στην αποστολή SPAM mail, στην αναβάθμιση των ίδιων των bots, στην αλλαγή C&C κ.ο.κ.

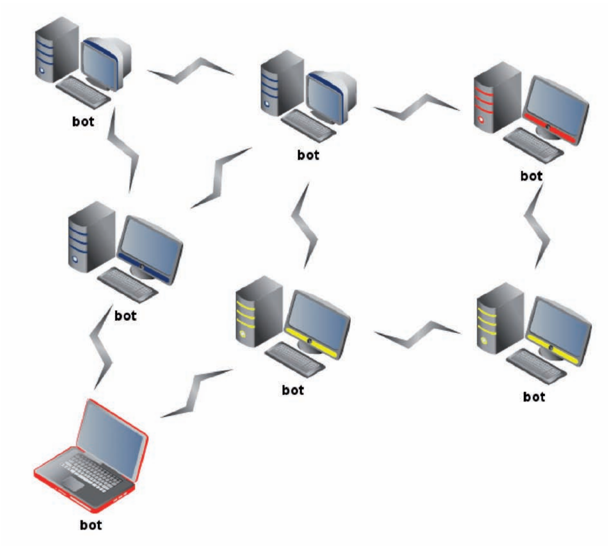
Χρησιμοποιώντας παραπάνω C&C servers, ο botmaster αποφεύγει το single point of failure. Επίσης, αν τα κέντρα έχουν κατανεμηθεί γεωγραφικά έξυπνα, η επικοινωνία γίνεται γρήγορα και αποτελεσματικά. Το θέμα μ' αυτόν τον τρόπο οργάνωσης του botnet είναι η σχετικά περίπλοκη διαχείριση και συντήρησή του.

Κύκλος ζωής ενός bot

Τα bots --και κατ’ επέκταση τα botnets-- κατά τη διάρκεια της ύπαρξής τους ακολουθούν συγκεκριμένα στάδια. Το πρώτο στάδιο ξεκινά με την εκμετάλλευση κάποιας αδυναμίας σε έναν υπολογιστή, ώστε να εγκατασταθεί ένα bot σ’ αυτόν και να γίνει μέρος του botnet. Η μόλυνση μπορεί να επιτευχθεί με πολλούς και διάφορους τρόπους, ενώ ο ίδιος ο υπολογιστής αφού γίνει πλέον zombie μπορεί να χρησιμοποιήσει κάποιους απ’ αυτούς, προκειμένου να διαδώσει το bot σε όσους άλλους υπολογιστές μπορεί!



Ένας τρόπος διάδοσης είναι μέσω κακόβουλου κώδικα. Ο δημιουρ-



Peer-to-Peer: Η πλέον εξελιγμένη αρχιτεκτονική για ένα botnet! Δεν υπάρχει κεντρική διαχείριση και κάθε bot παίρνει διάφορους ρόλους. Στα θετικά συγκαταλέγεται η ασφάλεια για τον botmaster. Μόνο αρνητικό είναι η περιστασιακή καθυστέρηση στη διάδοση των εντολών εντός του botnet.

γός/κάτοχος του bot επινοεί τρόπους ώστε να πείσει το θύμα να εκτελέσει αυτόν τον κώδικα στον υπολογιστή του. Μεταξύ αυτών των τρόπων είναι τα μολυσμένα mail attachments, τα websites που τρέχουν κώδικα στους browsers των επισκεπτών (π.χ., κάποιο Java applet), τα SPAM σε προγράμματα instant messaging που περιέχουν links του τύπου "click me to see something" κ.ο.κ.

Άλλος τρόπος για μόλυνση είναι μέσω αυτοματοποιημένων επιθέσεων που εκμεταλλεύονται συγκεκριμένες αδυναμίες (vulnerabilities). Για να μεγαλώνει αυτόματα ένα botnet, ο δημιουργός του bot ουσιαστικά έχει φροντίσει ώστε όλη η βρώμικη δουλειά να γίνεται από τα ήδη μολυσμένα μηχανήματα. Συγκεκριμένα, κάθε bot που έχει ήδη εγκατασταθεί κάνει ένα δικτυακό σκανάρισμα, ώστε να ανακαλύψει κι άλλους υπολογιστές με τους οποίους θα μπορούσε να επικοινωνήσει. Αρχικά εκτελεί ένα port scanning, για να δει ποιες θύρες των άλλων υποψήφιων θυμάτων είναι ανοικτές. Κατόπιν, δοκιμάζει ένα σετ γνωστών exploits ενάντια στις ανοικτές θύρες κάθε υπολογιστή. Θυμηθείτε ξανά το malware honeypot που είχαμε στήσει στο τεύχος 007 και πώς έπιανε συνεχώς variants του γνωστού worm Conficker, που εκμεταλλεύεται αδυναμίες της υπηρεσίας SMB στη θύρα 445 συστημάτων Windows, προκειμένου να εξαπλώνεται διαρκώς.

Τα botnets ψάχνουν για υπολογιστές με αδυναμίες, τις οποίες αν εντοπίσουν κι εκμεταλλευτούν είναι σε θέση να διαχειρίζονται απομακρυσμένα τα νέα θύματα. Αυτό το σύνολο αδυναμιών και αντίστοιχων exploits συνεχώς μεγαλώνει, καθώς είναι κοινό μυστικό ότι μετά από την έκδοση κάθε κρίσιμου patch από τη Microsoft πολλοί κακόβουλοι χρήστες ανά τον κόσμο αρχίζουν το reverse engineering: Σκοπός τους είναι ν' ανακαλύπτουν επακριβώς το κενό που καλύπτει το patch, καθώς και το πώς θα μπορούσαν να το εκμεταλλευτούν γράφοντας το ανάλογο exploit.

Τα botnets μεγαλώνουν κι εξαπλώνονται αφού είναι πάρα πολλοί εκείνοι οι χρήστες που αφήνουν τον υπολογιστή τους απροστάτευτο, μην εγκαθιστώντας αναβαθμίσεις, patches κι ενημερώσεις ασφαλείας. Μάλιστα ένα καλογραμμένο bot μπορεί να προσθέτει νέα exploits στο σύνολο των ήδη γνωστών. Εντολή για κάτι τέτοιο δίνει φυσικά ο botmaster, οπότε και κάθε zombie ουσιαστικά κατεβάζει τη νέα, εμπλουτισμένη έκδοση του bot.

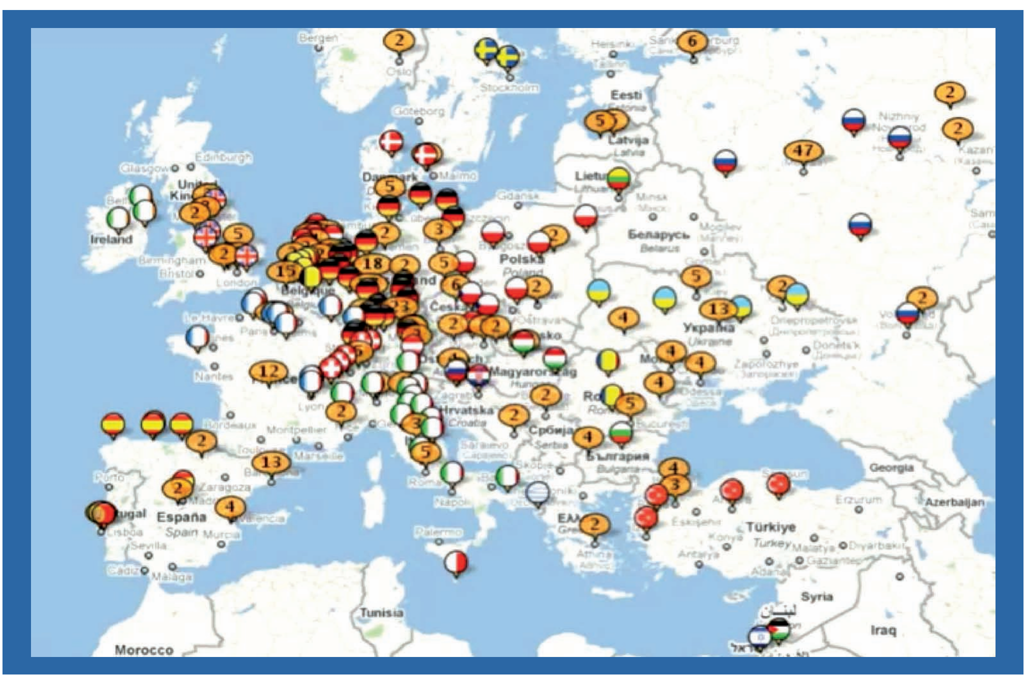
Ένας ακόμα τρόπος διάδοσης των bots είναι η αναζήτηση υπολογιστών με κάποιο backdoor που έχει ανοίξει από άλλο, ήδη εγκατεστημένο κακόβουλο λογισμικό. Για παράδειγμα, το γνωστό SDBot ψάχνει να βρει ήδη μολυσμένους υπολογιστές από RATs (Remote Access Trojans) ή worms όπως τα Kuang, MyDoom και SubSeven. Αυτά αφήνουν ως backdoor κάποια ανοικτή θύρα κι ένα bot μπορεί απλά να χρησιμοποιήσει τον default κωδικό σύνδεσης σ' αυτή, προκειμένου να καταλάβει τον υπολογιστή.

Αυτή η μέθοδος είναι πλέον ξεπερασμένη, αλλά μέχρι και πριν μερικά χρόνια χρησιμοποιούταν κανονικά!

Τέλος, ένα bot μπορεί να εκτελεί επιθέσεις brute force. Για παράδειγμα, αφού με κάποιον τρόπο εγκατασταθεί σε έναν υπολογιστή, ενδέχεται ν' αρχίσει brute force attack προς την υπηρεσία Windows File Sharing, ώστε να διαδοθεί σε άλλους υπολογιστές του τοπικού δικτύου. Για να το πετύχει δοκιμάζει μια ολόκληρη λίστα με usernames και passwords που έχει ήδη ενσωματωμένη στον κώδικά του -- ή κατεβάζει σχετική λίστα από το Διαδίκτυο. Κατ' αυτόν τον τρόπο, π.χ., λειτουργεί ένα άλλο γνωστό bot, με την ονομασία Rbot.

Αυτοί είναι --λίγο πολύ-- οι τρόποι διάδοσης και εξάπλωσης ενός bot, πάντα με στόχο τη διεύρυνση του αντίστοιχου botnet. Υπενθυμίζουμε ότι μελετούμε τα βήματα που αποτελούν τον κύκλο ζωής ενός bot. Το παραπάνω λοιπόν ήταν το πρώτο βήμα. Tow bot, αφού εγκατασταθεί σε έναν υπολογιστή, θα περάσει στο δεύτερο βήμα. Πρόκειται για το λεγόμενο "rallying" (συσπείρωση). Ουσιαστικά, το bot πραγματοποιεί την πρώτη του σύνδεση με το κέντρο ελέγχου C&C κι ανακοινώνει την παρουσία του. (Σ.τ.Ε. Φαντάζομαι κάτι σαν "this is bot delta-niner-zero, reporting for duty!», με χαρακτηριστική αμερικάνικη προφορά.) Υπάρχουν διάφοροι τρόποι (πρωτόκολλα) επικοινωνίας κι αρχιτεκτονικές οργάνωσης των botnets, τις οποίες θα δούμε παρακάτω. Αφού λοιπόν το bot γίνει ενεργό μέλος του botnet με αυτή του την ενέργεια, σειρά έχει το τρίτο βήμα που είναι κάποια ενδεχόμενη αναβάθμισή του. Κατά την αναβάθμιση, π.χ., το bot παίρνει ανανεωμένες λίστες διευθύνσεων κέντρων C&C, λαμβάνει νέα exploits, εξοπλίζεται με διάφορα scripts για την απενεργοποίηση antivirus γνωστών εταιρειών κ.ο.κ. Με αυτό τον τρόπο ο botmaster ουσιαστικά κάνει ό,τι μπορεί, ώστε το νέο zombie να παραμείνει στο botnet και συνεπώς υπό τον έλεγχό του!

Εδώ βλέπουμε έναν χάρτη με κέντρα C&C διαφόρων HTTP botnets, όπως έχουν καταγραφεί στην ευρύτερη περιοχή μας. Τα αριθμητικά πρωτεία κατέχει η Ρωσία, ενώ παρατηρούμε και δύο ταπεινά κέντρα στην Αθήνα που μας βάζουν σε σκέψεις!



Σταθμός καλεί βάση, όβερ

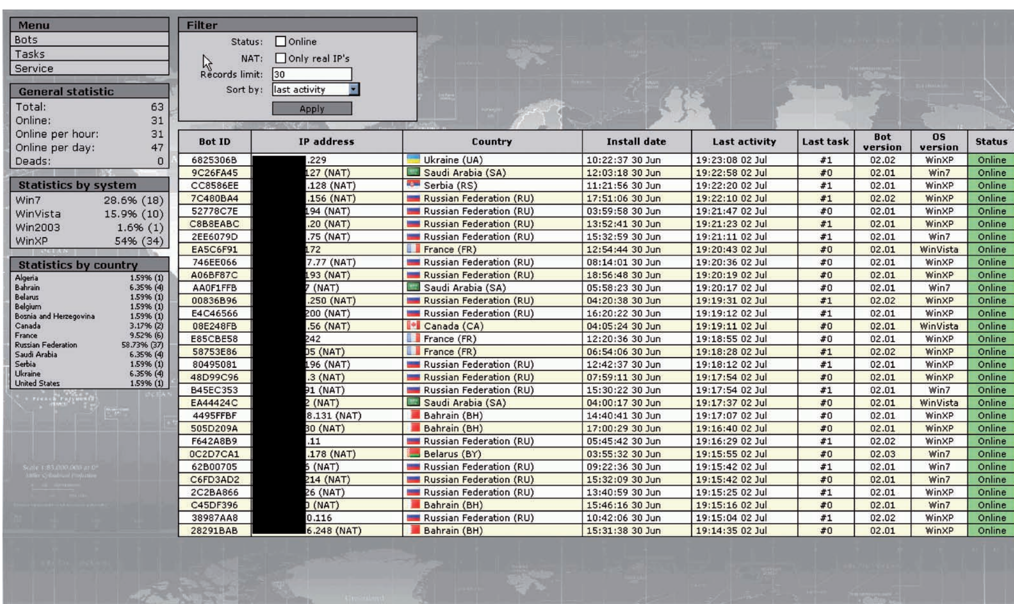
Όπως αναφέραμε, σε κάποια φάση το bot θα πρέπει να επικοινωνήσει με το κέντρο Command & Control όπου θα ανακοινώσει την παρουσία του και από εκεί θα δέχεται πλέον εντολές. Προκύπτει επομένως το ζήτημα της επικοινωνίας του botnet και του συγχρονισμού όλων των bots. Τα περισσότερα botnets βασίζονταν εδώ και μια δεκαετία στο πρωτόκολλο IRC. Φυσικά τα πράγματα εξελίσσονται πολύ γρήγορα, καθώς λαμβάνει χώρα ένας διαρκής αγώνας δρόμου μεταξύ κυβερνοεγκληματιών που χειρίζονται μεγάλα botnets και ειδικών ασφαλείας με την ιδιότητα των bothunters, οι οποίοι καταλαμβάνουν κι απενεργοποιούν όσα botnets μπουν στο στόχαστρο τους.

Ας επιστρέψουμε όμως στο πρωτόκολλο IRC. Γιατί είναι το αγαπημένο για τη δημιουργία botnets; Για τέσσερις απλούς λόγους:

1. Επιτρέπει αμφίδρομη επικοινωνία σε πραγματικό χρόνο μεταξύ server και client, όπου server στην περίπτωση μας είναι το κέντρο C&C και clients οι υπολογιστές zombies.
2. Είναι πολύ εύκολο να στηθεί ένας private IRC server -- ή ακόμα και να χρησιμοποιηθεί ένας απ' τους πολλούς υπάρχοντες.
3. Με έναν μόνο IRC server μπορείς να χειριστείς πολλά bots και πολλά botnets, καθώς σου παρέχει συνδέσεις με usernames και passwords, διάφορα κανάλια συνομιλίας, password-based authentication κ.ο.κ.
4. Μπορείς να πετύχεις απρόσκοπτη λειτουργία, συνδέοντας δύο και παραπάνω IRC servers μεταξύ τους.

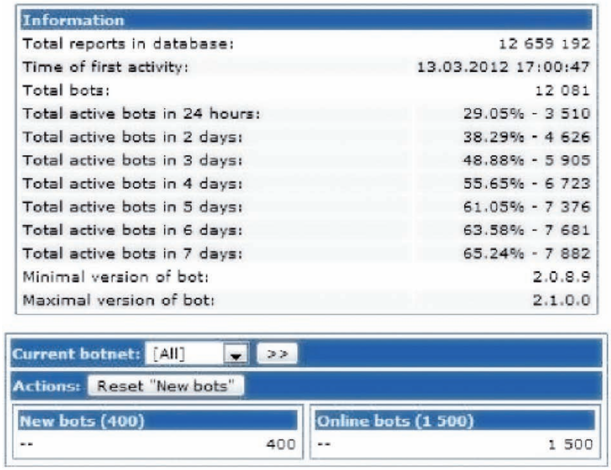
Παρόλα αυτά, ένα botnet που στηρίζεται στο IRC έχει μια πολύ συγκεκριμένη αδυναμία που προέρχεται από τη φύση του ίδιου του πρωτοκόλλου: Έχει αυτό που ονομάζουμε συγκεντρωτική (centralized) διαχείριση.

Το περιβάλλον διαχείρισης του Adromeda botnet, όπου έχουμε τη λίστα με όλα τα συνδεδεμένα bots και τις πληροφορίες τους. Απ' ό,τι φαίνεται, τα περισσότερα zombies έχουν εγκατεστημένα τα Windows XP. Όχι και πολύ μεγάλη έκπληξη, ε;



Bot ID	IP address	Country	Install date	Last activity	Last task	Bot version	OS version	Status
6825306B	[redacted]	Ukraine (UA)	10:22:37 30 Jun	19:23:08 02 Jul	#1	02.02	WinXP	Online
9C26FA45	[redacted]	Saudi Arabia (SA)	12:03:18 30 Jun	19:22:58 02 Jul	#0	02.01	Win7	Online
CC8586EE	[redacted]	Serbia (RS)	11:21:56 30 Jun	19:22:20 02 Jul	#1	02.01	WinXP	Online
7C480BA4	[redacted]	Russian Federation (RU)	17:51:06 30 Jun	19:22:10 02 Jul	#1	02.02	WinXP	Online
52778C7E	[redacted]	Russian Federation (RU)	03:59:58 30 Jun	19:21:47 02 Jul	#0	02.01	WinXP	Online
C8B8EABC	[redacted]	Russian Federation (RU)	13:52:41 30 Jun	19:21:23 02 Jul	#1	02.01	WinXP	Online
2EE6079D	[redacted]	Russian Federation (RU)	15:32:59 30 Jun	19:21:11 02 Jul	#1	02.01	Win7	Online
EA5C6F91	[redacted]	France (FR)	12:54:44 30 Jun	19:20:43 02 Jul	#0	02.01	WinVista	Online
746EE066	[redacted]	Russian Federation (RU)	08:14:01 30 Jun	19:20:36 02 Jul	#0	02.01	WinXP	Online
A06BF87C	[redacted]	Russian Federation (RU)	18:56:48 30 Jun	19:20:19 02 Jul	#0	02.01	WinXP	Online
AA0F1FFB	[redacted]	Saudi Arabia (SA)	05:58:23 30 Jun	19:20:17 02 Jul	#0	02.01	Win7	Online
00836B96	[redacted]	Russian Federation (RU)	04:20:38 30 Jun	19:19:31 02 Jul	#1	02.02	WinXP	Online
E4C46566	[redacted]	Russian Federation (RU)	16:20:22 30 Jun	19:19:12 02 Jul	#1	02.01	WinXP	Online
08E248FB	[redacted]	Canada (CA)	04:05:24 30 Jun	19:19:11 02 Jul	#0	02.01	WinVista	Online
E85CBE58	[redacted]	France (FR)	12:20:36 30 Jun	19:18:55 02 Jul	#0	02.01	WinXP	Online
58753E86	[redacted]	France (FR)	06:54:06 30 Jun	19:18:28 02 Jul	#1	02.02	WinXP	Online
80495081	[redacted]	Russian Federation (RU)	12:42:37 30 Jun	19:18:12 02 Jul	#1	02.01	WinXP	Online
48D99C96	[redacted]	Russian Federation (RU)	07:59:11 30 Jun	19:17:54 02 Jul	#0	02.01	WinXP	Online
B45EC353	[redacted]	Russian Federation (RU)	15:30:22 30 Jun	19:17:54 02 Jul	#1	02.01	Win7	Online
EA44424C	[redacted]	Saudi Arabia (SA)	04:00:17 30 Jun	19:17:37 02 Jul	#0	02.01	WinVista	Online
4495FFBF	[redacted]	Bahrain (BH)	14:40:41 30 Jun	19:17:07 02 Jul	#0	02.01	WinXP	Online
505D209A	[redacted]	Bahrain (BH)	17:00:29 30 Jun	19:16:40 02 Jul	#0	02.01	WinXP	Online
F642A8B9	[redacted]	Russian Federation (RU)	05:45:42 30 Jun	19:16:29 02 Jul	#1	02.02	WinXP	Online
0C2D7CA1	[redacted]	Belarus (BY)	03:55:32 30 Jun	19:15:55 02 Jul	#0	02.03	Win7	Online
62B00705	[redacted]	Russian Federation (RU)	09:22:36 30 Jun	19:15:42 02 Jul	#1	02.01	Win7	Online
C6FD3AD2	[redacted]	Russian Federation (RU)	15:32:09 30 Jun	19:15:42 02 Jul	#0	02.01	Win7	Online
2C2BA866	[redacted]	Russian Federation (RU)	13:40:59 30 Jun	19:15:25 02 Jul	#1	02.01	WinXP	Online
C45DF396	[redacted]	Bahrain (BH)	15:46:16 30 Jun	19:15:16 02 Jul	#0	02.01	Win7	Online
38987AA8	[redacted]	Russian Federation (RU)	10:42:06 30 Jun	19:15:04 02 Jul	#1	02.02	WinXP	Online
28291BAB	[redacted]	Bahrain (BH)	15:31:38 30 Jun	19:14:35 02 Jul	#0	02.01	WinXP	Online

Δηλαδή, όλα τα bots ακούνε για εντολές από ένα ή παραπάνω συγκεκριμένα κέντρα C&C. Αν λοιπόν μια ομάδα από bothunters καταφέρει και ανακαλύψει ποιός ή ποιοι είναι οι συγκεκριμένοι IRC servers που επιτελούν καθήκοντα C&C για ένα botnet, μπορούν πολύ απλά να το διαλύσουν. Αυτό γίνεται είτε ειδοποιώντας τον πάροχο/ISP, ο οποίος άθελα του ή μη παρέχει τη διαδικτυακή υποδομή στο botnet, είτε ακόμη κάνοντας takeover, δηλαδή καταλαμβάνοντας οι ίδιοι το botnet «κλέβοντάς» το από τον botmaster, με στόχο φυσικά την εξάρθρωση του botnet. Το τελευταίο μπορεί να μην αποτελεί την επιτομή της νομιμότητας ή της ηθικής, αλλά είναι κάτι που μπορεί και έχει γίνει, σίγουρα για καλό σκοπό. Εξάλλου, ποιός ακριβώς θα διαμαρτυρηθεί; :P



Μέρος από το control panel ενός botmaster. Ο συγκεκριμένος έχει τροποποιήσει το γνωστό Zeus bot κι έχει καταφέρει να εντάξει στο botnet που δημιούργησε περί τους 12.000 υπολογιστές! Σύμφωνα με τα λεγόμενα του, εκτός των άλλων έχει βάλει όλα τα zombies να κάνουν Bitcoin mining για τον ίδιο!

Ακόμη και στην περίπτωση που, όπως είπαμε, πολλοί IRC servers μαζί συγκροτούν τα κέντρα C&C, οι IP διευθύνσεις τους πρέπει να είναι ενσωματωμένες (hardcoded) στον κώδικα των bots, οπότε σε περιπτώσεις προβλημάτων στην επικοινωνία ο μόνος τρόπος για να ξέρουν πού να συνδεθούν είναι να παίρνουν από τον botmaster νέες λίστες με διευθύνσεις C&C. Αυτό φυσικά έχει επίσης τις αδυναμίες του, καθώς bothunters μπορούν να πιάσουν bots σε εργαστηριακό περιβάλλον και να βλέπουν οι ίδιοι όλες τις αλλαγές που πραγματοποιούνται στις IPs. Πολύ περισσότερο, μπορούν απλά να παρεμβάλλονται στο κανάλι επικοινωνίας και να παρακολουθούν σιωπηρά τι συμβαίνει.

Η λύση που εφάρμοσαν οι απανταχού κακόβουλοι προγραμματιστές των bots, ήταν να χρησιμοποιήσουν την τεχνολογία DNS (Domain Name System). Το DNS, όπως γνωρίζετε, επιτελεί μια απλή διεργασία: αντιστοιχίζει domain names σε διευθύνσεις IP. Με αυτήν τη λογική, τα bots δεν συνδέονται πλέον σε κάποιες συγκεκριμένες hardcoded διευθύνσεις IP αλλά σε κάποιο domain, όπως, π.χ., το controlcenter.botnet.com. Το domain αυτό μπορεί να αντιστοιχεί κάθε φορά σε όποια IP επιλέξει ο botmaster. Οπότε και να "πέφτουν" κάποιοι από τους IRC servers, απλά αλλάζει η IP στην οποία αντιστοιχεί το domain και το botnet συνεχίζει να λειτουργεί. Βάλτε στην εικόνα πολλά και διάφορα domains κατοχυρωμένα μέσω διαφόρων εταιρειών σε διαφορετικές χώρες, και η κατάσταση γίνεται αρκετά περίπλοκη. Προσθέστε στο μείγμα και την εφαρμογή multihoming στα DNS records, δηλαδή ένα domain να αντιστοιχεί σε παραπάνω από μία IPs, οπότε και να "πέσει" κάποια απαντούν οι άλλες! Ειδικά η τελευταία αυτή τεχνική χρησιμοποιείται για να μπορούν τα bots να βρίσκουν πάντα ένα από τα υπάρχοντα κέντρα C&C. Στην απλή εκδοχή της ονομάζεται single-flux και είναι στην ουσία η αντιστοίχιση εκατοντάδων ή ακόμα και χιλιάδων διευθύνσεων IP με ένα domain name. Οι διευθύσεις αυτές αλλάζουν διαρκώς και γρήγορα -- κι αυτό ονομάζεται γενικά fast-flux. Έτσι, το ίδιο domain name δείχνει κάθε φορά σε μία από τις πολλές διαθέσιμες IP. Στην προχωρημένη εκδοχή της μεθόδου, που ονομάζεται double-flux, εκτός από τις IPs στις οποίες δείχνουν τα domain

names (A records) αλλαγές έχουμε και στις διευθύνσεις IP των ίδιων των DNS servers (NS records), οι οποίοι φροντίζουν για τις ίδιες τις αντιστοιχίσεις domain name <--> IP(s). Προφανώς, όλες αυτές οι υπέροχες επινοήσεις αποτελούν έναν πρώτης τάξεως πονοκέφαλο για τους bothunters και τις εταιρείας ασφαλείας.

HTTP, P2P και η νέα γενιά των botnets

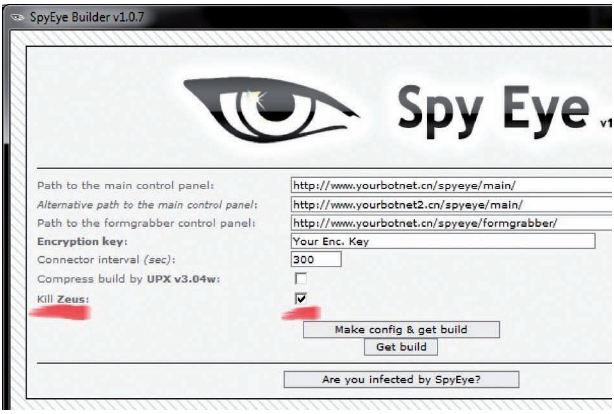
Πέρα από τα παραδοσιακά IRC-based botnets έχουμε και πιο εξελιγμένες μορφές καθώς οι προγραμματιστές των bots δουλεύουν σκληρά, αφού είναι πολλά τα λεφτά :) Έτσι, υπάρχουν και δύο νέες μορφές botnets που αυτή τη στιγμή είναι στο προσκήνιο: τα HTTP ή Web-based botnets, καθώς και τα P2P botnets.

Τα HTTP botnets είναι στην ουσία παρόμοια στη φιλοσοφία με τα IRC botnets, αλλά με δύο σημαντικές διαφορές. Η πρώτη είναι πως ο έλεγχος δεν γίνεται πλέον μέσω του πρωτοκόλλου IRC αλλά μέσω ενός HTTP κι ενός εύχρηστου web interface. Αυτό το τελευταίο σημαίνει ότι και κάποιος χωρίς ιδιαίτερες γνώσεις θα μπορούσε να χειριστεί το botnet με σχετική ευκολία. Κι αυτό είναι εξαιρετικά σημαντικό, αφού το λογισμικό δημιουργίας των botnets πωλείται με λεγόμενα DIY (do-it-yourself) botnet kits. Ο καθένας, δηλαδή, θα μπορούσε, πηγαίνοντας σε διάφορα σκιώδη fora, να αγοράσει ένα τέτοιο kit και να φτιάξει ένα δικό του botnet, το οποίο θα μπορούσε σχετικά εύκολα να χειριστεί. Αν ακόμη κι αυτό του φαίνεται δύσκολο, θα μπορούσε απλά να νοικιάσει ένα botnet ή ένα κομμάτι αυτού και να το χειριστεί απευθείας, μέσω του web interface που θα σηκώσει για αυτόν ο αρχικός κάτοχος. Αυτή είναι η πρώτη διαφορά με τα IRC botnets.

Η δεύτερη έχει να κάνει με τον τρόπο επικοινωνίας μεταξύ των bots και του κέντρου C&C. Στην περίπτωση του IRC, τα bots είναι συνεχώς συνδεδεμένα στον server που παίζει ρόλο C&C και δέχονται εντολές από αυτόν με την λογική του "push". Δηλαδή τη στιγμή που μια νέα εντολή δίνεται από τον botmaster στο κέντρο C&C αυτό αναλαμβάνει και την προωθεί σε όλα τα bots, που με τη σειρά τους τη λαμβάνουν και την εκτελούν. Στα HTTP botnets, η επικοινωνία γίνεται με τη λογική του "pull". Τα bots δεν είναι συνεχώς συνδεδεμένα με το κέντρο C&C, αλλά ανά προκαθορισμένα χρονικά διαστήματα κάνουν μια σύνδεση με αυτό για να δουν αν υπάρχουν νέες εντολές, και αν ναι τις ζητούν και τις λαμβάνουν. Επειδή η επικοινωνία μέσω HTTP είναι πολύ κοινή και συνηθισμένη, όλο αυτό μπορεί να περάσει αρκετά απαρατήρητο. Για να το σταματήσεις θα πρέπει να μπλοκάρεις όλη τη διαδικτυακή σύνδεση, πράγμα καθόλου πρακτικό για απλούς χρήστες, εταιρείες κ.λπ. που έχουν πέσει θύματα. Μάλιστα εκτός από το HTTP μπορεί να χρησιμοποιηθεί και το πρωτόκολλο HTTPS, όπου ο μηχανισμός επικοινωνίας παραμένει ο ίδιος αλλά πλέον η διαδικτυακή κίνηση είναι κρυπτογραφημένη!

Τα δε P2P (peer-to-peer) botnets αποτελούν το πιο εξελιγμένο τεχνολογικό επίτευγμα των προγραμματιστών bots. Ο μηχανισμός επικοινωνίας P2P είναι το ακριβώς αντίθετο του συγκεντρωτισμού, κι αυτό συνεπάγεται πολλά θετικά -- για το botnet. Τα bots σχηματίζουν ένα δίκτυο σαν πλέγμα και χωρίς ιεραρχία, η διαχείριση του οποίου είναι εντελώς κατανεμημένη. Αυτό σημαίνει πως δεν υπάρχει ένα ή περισσότερα συγκεκριμένα κέντρα C&C. Κάθε υπολογιστής που ανήκει στο botnet μπορεί να παίξει αυτό το ρόλο, όπως και διάφορους άλλους. Προφανώς πρόκειται για εξαιρετικά σημαντικό χαρακτηριστικό, καθώς τώρα δεν υπάρχει κάποια υποδομή ενάντια στην οποία μπορείς να κινηθείς για να σταματήσεις το

botnet. Καθένας από τους χιλιάδες μολυσμένους υπολογιστές μπορεί τη μια στιγμή να είναι zombie ενώ την επόμενη να έχει μετατραπεί σε κέντρο C&C. Ένας botmaster θα μπορούσε π.χ. να συνδεθεί σε δύο από αυτούς στη σειρά, χρησιμοποιώντας τους ως proxies για να κρύψει την ταυτότητα του, και κατόπιν σε έναν τρίτο, στον οποίο θα έδινε τις εντολές που θέλει. Μέσω αυτού θα μεταφέρονταν σταδιακά, από zombie σε zombie, σε όλο το δίκτυο-πλέγμα. Την επόμενη φορά, ο botmaster θα διάλεγε μια άλλη, άσχετη διαδρομή. Να σημειώσουμε εδώ ότι τέτοιες τεχνικές επικοινωνίας και διαχείρισης χρησιμοποιούν τα πλέον εξελιγμένα bots.



Ο builder του EyeSpy bot. Όπως βλέπουμε, το συγκεκριμένο bot επιτίθεται ενάντια στο Zeus! Κάθε δημιουργός bot έχει να αντιμετωπίσει κι όλους τους άλλους "συναδέλφους" του. Έχε τους φίλους σου κοντά και τους εχθρούς σου ακόμα πιο κοντά, όπως λέει και το ρητό.

Υπάρχουν περιπτώσεις όπου, λόγω κάποιας αδυναμίας του ίδιου του λογισμικού ενός P2P botnet, ελοχεύει ο κίνδυνος της διάλυσής του. Μια κλασική τέτοια περίπτωση αποτέλεσε το Storm botnet. Το συγκεκριμένο botnet, το οποίο κάποτε είχε υπό τον έλεγχο του εκατοντάδες χιλιάδες υπολογιστές, παρουσίασε μια αδυναμία την οποία κάποιοι ερευνητές μπόρεσαν και εκμεταλλεύτηκαν. Έγραψαν έτσι ένα καλοπροαίρετο worm, το οποίο εισερχόταν σε υπολογιστές zombies, κατέβαζε ένα αρχείο για να τους καθαρίσει από το Stormbot και κατόπιν εξαπλωνόταν σε άλλους υπολογιστές του botnet (χάρη στην επικοινωνία P2P). Η όλη επιχείρηση και το αρχείο καθαρισμού πήρε την όχι και τόσο εμπνευσμένη ονομασία "Stormfucker" και παρουσιάστηκε στο 25ο Chaos Communication Congress. Λόγο νομικών περιορισμών δεν ήταν δυνατή η εφαρμογή της λύσης σε μαζική κλίμακα, καθώς η εγκατάσταση κάποιου προγράμματος χωρίς τη θέληση του χρήστη και η εκτέλεση λειτουργιών στον υπολογιστή του είναι παράνομη ενέργεια. Η Microsoft τελικά έδωσε μια μερική λύση μέσω του Malicious Software Removal Tool, το οποίο σύμφωνα με την εταιρεία καθάρισε πάνω από 500.000 υπολογιστές συνολικά.

Η νέα γενιά των botnets παρουσιάζει και άλλα ενδιαφέροντα χαρακτηριστικά. Κάποια από αυτά είναι η χρήση διαφόρων τεχνικών κρυπτογράφησης της επικοινωνίας και των δεδομένων που ανταλλάσσονται μεταξύ των bots, όπως και η χρήση custom packers για τη δημιουργία και τη συμπίεση των εκτελέσιμων αρχείων ώστε να είναι δύσκολο το reverse engineering τους. Επίσης αρκετά botnets έχουν την ικανότητα αυτοπροστασίας και αν γίνει αντιληπτό ότι κάποιος τα σκανάρει ή γενικά προσπαθεί κάτι να ανιχνεύσει, τότε αρχίζουν επίθεση DDoS ενάντια στις διευθύνσεις από όπου αυτές οι ενέργειες προέρχονται. Τέλος υπάρχουν και bots που αντιλαμβάνονται αν εκτελούνται σε πραγματικό ή εικονικό περιβάλλον (σε virtual machine) και αν συμβαίνει το δεύτερο τότε αποκτούν μια εντελώς διαφορετική συμπεριφορά με σκοπό να μπερδέψουν αυτόν που προσπαθεί να τα αναλύσει.

Show me the money

Μετά απ' όλα τα προηγούμενα, εύλογα κανείς αναρωτιέται: προς τι όλη η φασαρία; Τι έχουν τέλος πάντων να κερδίσουν οι προγραμματιστές των bots ή ο εκάστοτε botmaster; Είναι ώρα να δούμε πότε χρησιμοποιούνται

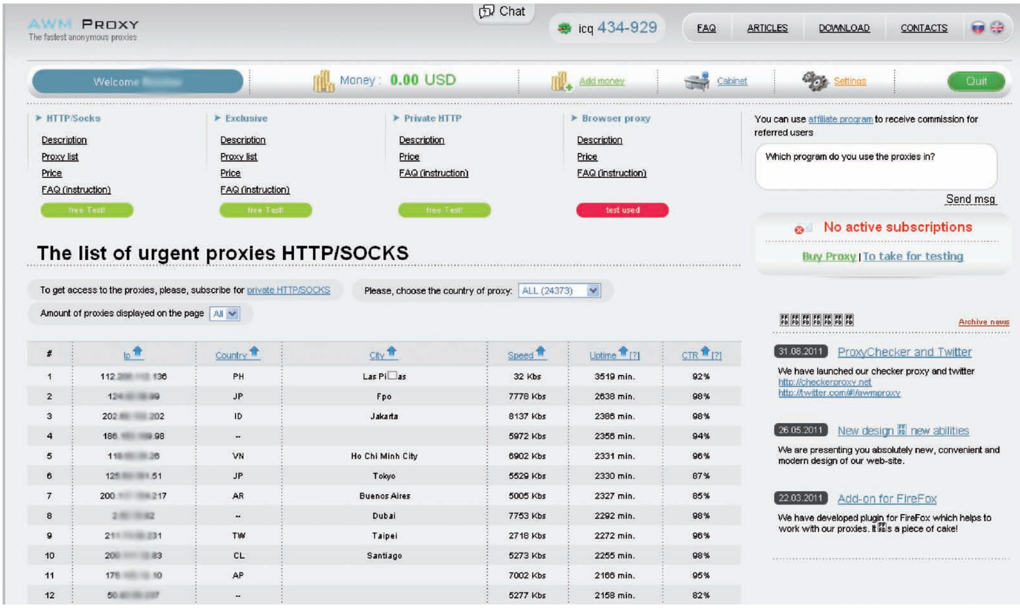
τα botnets και πώς επιφέρουν χρήματα σε όλους αυτούς που είναι ανακατεμένοι με την ύπαρξη και λειτουργία τους.

Ας αρχίσουμε με τους προγραμματιστές και δημιουργούς των bots, οι οποίοι είναι μια ξεχωριστή κατηγορία από μόνοι τους. Αυτό γιατί προγραμματιστές και χειριστές των botnets δεν είναι συνήθως τα ίδια πρόσωπα. Οι προγραμματιστές των bots είναι συχνά ευφυή άτομα με καλές γνώσεις διαδικτυακού προγραμματισμού, που για δικούς τους λόγους έχουν αναμειχθεί σε αυτή την παράνομη αγορά. Ένας φυσικά από τους πιο ισχυρούς λόγους είναι το κέρδος. Τα εξελιγμένα bots συνήθως είναι προς πώληση σε κάθε ενδιαφερόμενο. Αγγελίες για αυτά μπαίνουν σε διάφορα fora, τα οποία κινούνται εκτός ή το πολύ στα όρια της νομιμότητας. Τα λεφτά δεν είναι λίγα. Για παράδειγμα, ένα νέο λογισμικό botnet με την ονομασία THOR, το οποίο κάνει χρήση της τεχνολογίας P2P και σύμφωνα με τους δημιουργούς του πλησιάζει στην ολοκλήρωση, έχει αρχική τιμή πώλησης 8.000 δολλάρια στην underground αγορά. Υπάρχουν και οικονομικότερες λύσεις, βέβαια, όπως το Citadel: Αποτελεί κλώνο του πολύ γνωστού Zeus bot και διατίθεται στην προνομιακή τιμή των 2.400 δολαρίων, με πρόσθετες χρεώσεις για έξτρα modules που μπορούν να ενσωματωθούν σε αυτό. Από την άλλη υπάρχει και το απλό Andromeda bot, το οποίο μπορεί να αγοραστεί για 200 δολλάρια.

Στις εξερευνήσεις μας με έκπληξη ανακαλύψαμε κι ένα πακέτο με την ονομασία Malware Guard, το οποίο ενσωματώνει iptables rules που μπλοκάρουν οκτώ περίπου εκατομμύρια διευθύνσεις IP από κυβερνητικούς οργανισμούς, ερευνητικά εργαστήρια, πανεπιστήμια, εταιρείες ασφαλείας και γενικά οποιονδήποτε θα είχε όφελος να «καταστρέψει» ένα botnet. Με το Malware Guard θα μπορούσαν να ασφαλιστούν οι servers που λειτουργούν, π.χ., ως κέντρα C&C. Κι επειδή τίποτε δεν είναι δωρεάν σ' αυτόν το χώρο, το πακέτο κοστίζει 250 δολλάρια για κάθε server.

Ας δούμε τώρα τι χρήσεις έχει ένα τυπικό botnet και τι οφέλη έχει ο εκάστοτε botmaster. Πρώτη περίπτωση είναι η διενέργεια επιθέσεων DDoS. Ο botmaster δίνει εντολή και χιλιάδες υπολογιστές κατακλύζουν με αιτήσεις συγκεκριμένους web servers, καθιστώντας τους στην ουσία μη λειτουργικούς. Πολλές εταιρείες δέχονται απειλές από ιδιοκτήτες botnets. Ένα παράδειγμα αποτελεί το mybet.com. Οι ιδιοκτήτες του δέχτηκαν ένα σημείωμα κατά τη διάρκεια του EURO 2004, στο οποίο οι εκβιαστές ζητούνταν 15.000 δολλάρια προκειμένου να μην επιτεθούν στο site. Όταν τα λεφτά δεν παραδόθηκαν το site όντως έπεσε, λυγίζοντας υπό το βάρος της επίθεσης DDoS που ακολούθησε. Σύμφωνα με τον μη κερδοσκοπικό οργανισμό Shadowserver, που ασχολείται με θέματα σχετικά με malware και botnets, το 2008 οι κυβερνοεγκληματίες χειριστές διαφόρων botnets έβγαλαν συνολικά 20 εκατομμύρια δολλάρια από επιθέσεις DDoS.

Μια άλλη χρήση των botnets είναι η υποκλοπή κωδικών και στοιχείων από τους χρήστες-θύματα, μέσω διαφόρων τεχνικών όπως το keylogging. Αγαπημένα στοιχεία είναι οι διευθύνσεις email και οι συνδυασμοί username-password για υπηρεσίες που έχουν σχέση με χρηματικές συναλλαγές, όπως Paypal, online banking κ.α. Μια λίστα με ένα εκατομμύριο διευθύνσεις email κοστίζει από 20 ως 100 δολλάρια στη μαύρη αγορά, ενώ στοιχεία σύνδεσης σε τραπεζικούς λογαριασμούς και σχετικές υπηρεσίες πωλούνται και μεμονωμένα. Οι ίδιοι οι χειριστές των botnets, φυσικά, είναι οι πρώτοι που κλέβουν ποσά από τους λογαριασμούς των θυμάτων τους. Υπολογίζεται ότι από τη δράση μιας μόνο ομάδας από την Ανατολική Ευρώπη, η οποία διατηρούσε ένα botnet βασισμένο στο Zeus, εκλάπησαν



180 εκατομμύρια δολλάρια το 2010 από τα οποία τα 70 δεν έχουν επιστραφεί. Η συγκεκριμένη ομάδα πάντως διαλύθηκε και περίπου 65 άτομα σε Αμερική, Αγγλία και Ουκρανία συνελήφθησαν.

Το συγκεκριμένο κατάστημα πουλάει συνδρομές σε anonymous proxies. Μόνο που αυτοί είναι στην ουσία τα zombies ενός botnet, μέσω των οποίων κάθε αγοραστής μπορεί να σερφάρει κρύβοντας την ταυτότητα του! Μάλιστα οι δημιουργοί του έχουν φτιάξει και συνοδευτικό Firefox plugin, για μεγαλύτερη ευκολία στην εναλλαγή.

Η πιο χαρακτηριστική εφαρμογή των botnets θα λέγαμε ότι είναι η αποστολή SPAM. Εκατομμύρια διαφημιστικών mail στέλνονται καθημερινά και υπολογίζεται ότι το 80% αυτών προέρχεται από υπολογιστές zombies. Το Storm botnet, που αναφέρθηκε παραπάνω, έβγαζε για τους διαχειριστές του περίπου 7000 δολλάρια ημερησίως, πουλώντας διάφορα προϊόντα (κυρίως φαρμακευτικής και ερωτικής φύσεως) μέσω SPAM. Τα λεφτά αυτά προέρχονται από δύο κατηγορίες πληρωμών. Μία είναι η απευθείας κατάθεση ενός ποσού στους ιδιοκτήτες του botnet, για την αποστολή συγκεκριμένων SPAM mail. Άλλη είναι η πληρωμή των botmasters με ποσοστά επί των αγορών που πραγματοποιούνται -- ενώ μπορεί φυσικά να γίνει και συνδυασμός των δύο.

Μια άλλη παράνομη δραστηριότητα που σχετίζεται με τα botnets είναι το phishing. Πρόκειται για παραπλανητικά mails που στέλνονται κατά κύριο λόγο μαζικά --αλλά ορισμένες φορές και στοχευμένα-- με σκοπό να ξεγελάσουν τον παραλήπτη πείθοντας τον ότι προέρχονται από κάποια σημαντική πηγή όπως μία τράπεζα ή κάποιος άλλος οργανισμός. Σε ένα τέτοιο mail είτε ζητούνται από το χρήστη απευθείας κάποια προσωπικά στοιχεία, είτε δίνονται links προς ψεύτικες σελίδες που μοιάζουν με τις πραγματικές και καλούν τον επισκέπτη να κάνει login, υποκλέπτοντας έτσι τα στοιχεία της σύνδεσής του. Αυτές οι ψεύτικες ιστοσελίδες παράγονται μαζικά αλλά υπάρχει το θέμα του κλεισίματός τους, μόλις κάποιος τις ανακαλύψει. Γι' αυτό το λόγο βολεύει τους εκάστοτε παρανόμους να χρησιμοποιούν την τεχνολογία fast-flux των botnets, που περιγράψαμε παραπάνω, ώστε ακόμη κι αν οι servers που τις φιλοξενούν "πέφτουν", άλλοι να παίρνουν τη θέση τους χωρίς να χρειάζονται αλλαγές στα domain names κ.λπ. Οι δυνατότητες ενοικιάζονται, με το αζημίωτο φυσικά. Οι τιμές είναι από 1000 ως 2000 δολλάρια το μήνα.

Ένα κοινό πρόβλημα που αντιμετωπίζουν οι χρήστες-θύματα υπολογιστών zombies, είναι κάτι περίεργα pop-ups και μυστήριες διαφημίσεις

που ξεπετάγονται στα καλά του καθουμένου. Αυτό μόνο τυχαίο δεν είναι, καθώς μια άλλη αγαπημένη χρήση ενός botnet είναι η εγκατάσταση και ο διαμοιρασμός προγραμμάτων adware και spyware. Πολλές εταιρείες που ασχολούνται με το παράνομο διαδικτυακό μάρκετινγκ, πληρώνουν χρήματα για κάθε εγκατάσταση του κακόβουλου λογισμικού τους. Θα είχε ενδιαφέρον να παρακολουθήσετε ένα βίντεο από το γνωστό συνέδριο ασφαλείας DEF CON, όπου ένας πρώην προγραμματιστής adware και spyware εξηγεί πώς ακριβώς κατασκευάζονται και τι ενέργειες επιτελούν. Μπορείτε να το βρείτε εδώ:

`http://bit.ly/defcon18spydev`

Τέλος, όπως (σχεδόν) κάθε αγαθό Εκεί Έξω (TM), ένα botnet μπορεί να πωληθεί ή να ενοικιαστεί σε ενδιαφερόμενους. Για παράδειγμα, ένα botnet με σκοπό την αποστολή SPAM με ταχύτητα γύρω στα 1000 mail το λεπτό, μπορεί να ενοικιαστεί με 2000 δολλάρια το μήνα. Άλλο ένα παράδειγμα είναι η αγγελία για την πώληση του Shadow botnet προς 34.000 δολλάρια, το οποίο αποτελούταν από 100000 zombies και διαχειριζόταν από έναν δεκαεννιάχρονο Ολλανδό. Τι να πει κανείς, υπάρχουν χόμπι και υπάρχουν και χόμπι :)

Κύκλος ζωής των botnets

Κλείνοντας θα ήταν καλό να γίνει μια αναφορά στην πορεία κάθε botnet, αφού αναφερθήκαμε παραπάνω στον κύκλο ζωής των μεμονωμένων bots. Βλέποντας λοιπόν τη «μεγάλη εικόνα», ένα botnet ακολουθεί μια συγκεκριμένη πορεία πέντε φάσεων.

Πρώτη φάση είναι η κατασκευή του botnet. Σε αυτό το στάδιο συγγράφεται το bot, στήνονται τα κέντρα C&C και μετά το αφήνουν να εξαπλωθεί στα ανυποψίαστα θύματα. **Δεύτερη φάση** αποτελεί η ανίχνευση του. Όταν ένα botnet αρχίσει να μεγαλώνει και είναι ενεργό, τραβάει γρήγορα την προσοχή των bothunters που το βάζουν στο στόχαστρο τους. Περνάμε έτσι στην **τρίτη φάση** που είναι η εκούσια μόλυνση με το bot, προκειμένου να μελετηθεί διεξοδικά. Η μελέτη αυτή καθ' αυτή αποτελεί την επόμενη, **τέταρτη φάση** και γίνεται από malware experts, ώστε να καταστεί σαφές πώς λειτουργεί το bot εσωτερικά και τι διεργασίες επιτελεί. Ακολούθως περνάμε στην τελευταία, **πέμπτη φάση**, που είναι η τελική διάλυση του botnet μέσω διαφόρων μεθόδων -- ανάλογα με την αρχιτεκτονική του.

Πρέπει να ξέρετε ότι οι πέντε παραπάνω φάσεις μπορεί να καθυστερούν σε κάποιο σημείο, αλλά αργά ή γρήγορα πραγματοποιούνται. Αυτό το λέμε γιατί είμαστε σίγουροι πως όλο το κεφάλαιο botnets σας φάνηκε αρκετά ενδιαφέρον. Δεν πρέπει να ξεχνάμε ότι, ενώ σίγουρα αποτελούν ένα ενδιαφέρον και αρκετά επικερδές σπορ, η ενασχόληση με αυτό γίνεται πολύ επικίνδυνη μόλις κάποιος ξεφύγει από τα καθιερωμένα πλαίσια νομιμότητας. Μολαταύτα έχουμε δικαίωμα να μελετούμε τα botnets και να τα αναλύουμε, αν μη τι άλλο για να πλουτίζουμε τις γνώσεις μας.

Αυτό ακριβώς σκοπεύουμε να κάνουμε με το άρθρο που αρχίζει από τη σελίδα 68 του τεύχους που κρατάτε στα χέρια σας!

Εσείς μάθατε για την **προσφορά** Summer 2012;!

Μετά την **απίστευτη υποδοχή** που γνώρισε η προσφορά Ιουνίου, κι έπειτα από την κλήρωση 5 συνδρομών εφ όρου ζωής σε 5 τυχερούς αναγνώστες, επεκτείνουμε την προσφορά για **όλο το καλοκαίρι!**

Αγοράστε τώρα μια 12μηνη ή 24μηνη συνδρομή στο deltaHacker ή προσθέστε στη συνδρομή που έχετε άλλη μια 12μηνη ή 24μηνη συνδρομή, και αποκτήστε αυτόματα **5% έκπτωση** επί της κανονικής τιμής!

Επιπλέον, όσοι επιλέξετε 24μηνη συνδρομή μπαίνετε και σε κλήρωση που θα γίνει στις 31/8/2012, για πέντε (5) **συνδρομές εφ' όρου ζωής!** Ναι, σωστά διαβάσατε: Οι πέντε τυχεροί που θα κληρωθούν θα παίρνουν το περιοδικό για πάντα!

Για να εκμεταλλευτείτε την προσφορά **Summer 2012** αγοράστε την 12μηνη ή 24μηνη συνδρομή σας **έως και τις 26/8/2012**, συμπληρώνοντας τη φόρμα παραγγελιών στο

http://deltahacker.gr/order

Προσοχή: ακολουθούν ψιλά γράμματα!

Η έκπτωση του 5% και η κλήρωση για τις πέντε συνδρομές εφ' ότου ζωής, ισχύουν μόνον για όσους αγοράσουν 12μηνη ή 24μηνη συνδρομή **έως και τις 26/8/2012**.

Skill: Beginner

Tags: Canon Hacking Development Kit, CHDK, Digital Imaging Core, DigIC, firmware update

Hack your compact digital camera!

Δεν θα ήταν όμορφα αν είχαμε όλες τις ευκολίες μιας dSLR, σε μια μικρή, φτηνή φωτογραφική compact; Αν συνέβαινε κάτι τέτοιο, θα μπορούσαμε να πηγαίνουμε διακοπές χωρίς να εγκαταλείπουμε τις πλούσιες δυνατότητες που προσφέρουν οι άβολες στη μετακίνηση dSLR. Επίσης, θα μπορούσαμε να αναβαθμίσουμε εκείνη την παλιά compact, που έχουμε παρατήσει προ πολλού...

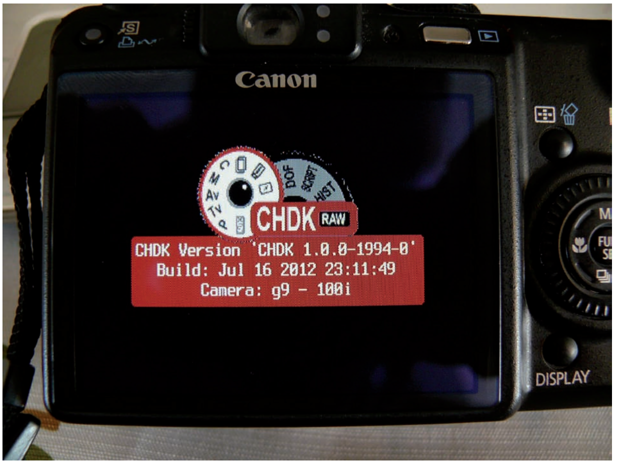


by Dr@G

Καλοκαιράκι! Όλο και κάποια κοντινή παραλία υπάρχει και, παρά την οικονομική κρίση, είναι σίγουρο ότι θα πεταχτούμε για μερικά μπανάκια, θα χαλαρώσουμε και θα ξεφύγουμε από την καθημερινότητα. Τώρα, θα μου πείτε ότι κάποιοι προτιμούν τα βουνά κι ότι όσο πιο ψηλά ανεβαίνουν, τόσο περισσότερο χαλαρώνουν και ηρεμούν. Υπάρχει κι αυτός ο κόσμος, πράγματι, αλλά αυτό δεν αναιρεί τη γενικότερη αλήθεια, όπως την περιέγραψε κι ο αγαπητός @subZraw: Όλοι θέλουμε να φορτίσουμε τις μπαταρίες μας και θα το κάνουμε, έστω κι αν αυτό προϋποθέτει να πάμε από το Α στο Β! Πάντως, αν και είναι μάλλον αδύνατο να συμφωνήσουμε στον ιδανικό τόπο διακοπών, είναι σίγουρο ότι συμφωνούμε στο τι θέλουμε να πάρουμε μαζί μας: Ανεξάρτητα λοιπόν από το αν μας αρέσουν οι θάλασσες ή τα βουνά, είναι βέβαιο ότι θα πάρουμε μαζί μας μια φωτογραφική μηχανή!

Ο γράφων έχει μεγάλη τρέλα με τη φωτογραφία. Του αρέσει να φωτογραφίζει τα πάντα, όπου βρεθεί κι όπου σταθεί. Θέλει να αποτυπώνει την ομορφιά της στιγμής, της φύσης, του μικρόκοσμου και πάει λέγοντας[1]. Κατά πάσα πιθανότητα, την ίδια τρέλα θα κουβαλούν και πολλοί ακόμα φίλοι του περιοδικού. Ωστόσο, λίγοι κουβαλάνε στις διακοπές τους μια dSLR. Σχεδόν όλοι μας, είτε έχουμε το μικρόβιο της φωτογραφίας είτε όχι, κουβαλάμε μια μικρή και βολική compact. Αλλά ξέρετε κάτι; Οι δυνατότητες αυτής της compact ενδέχεται να 'ναι πολλές περισσότερες, απ' όσες θέλει να μας κάνει να πιστέψουμε ο κατασκευαστής...

Σε αυτό το άρθρο θα ασχοληθούμε με την αναβάθμιση μερικών δημοφιλών φωτογραφικών μηχανών. Συγκεκριμένα, θα δούμε πώς μπορούμε να "εμπλουτίσουμε" το firmware στις Canon Powershot, ώστε ν' αποκτήσουν νέες, ενδιαφέρουσες ικανότητες! Για το σκοπό αυτό θα παίξουμε με ένα πακέτο λογισμικού που ονομάζεται Canon Hacking Development Kit (CHDK).



Η έκδοση του CHDK για το firmware GM1.00H, της Canon Powershot G9.

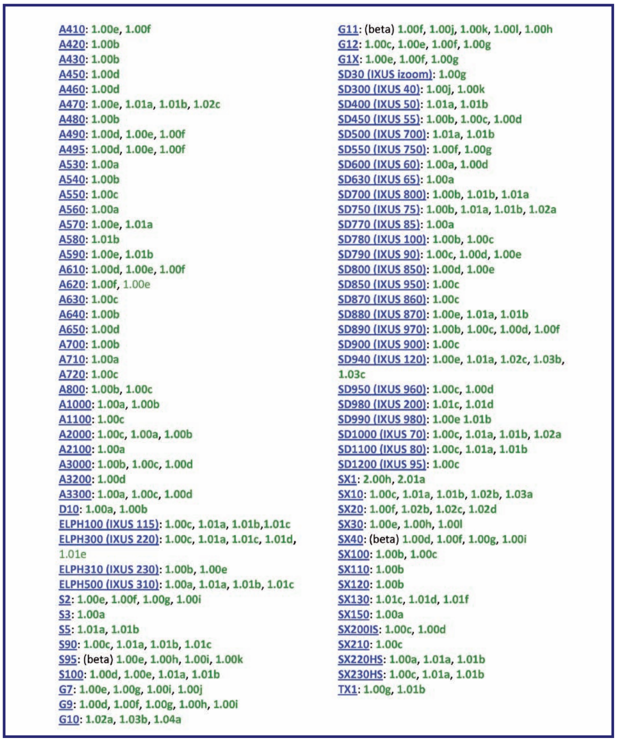
Γνωριμία με το CHDK

Δεν χρειάζεται να το λέμε συχνά, γιατί είναι κοινό χαρακτηριστικό *όλων* των φίλων του περιοδικού: Σε κανέναν από μας δεν αρέσουν οι περιορισμοί[2]! Κάπως έτσι, λοιπόν, τίθεται ένα εύλογο ερώτημα: Γιατί να μην έχουν οι μικρές φωτογραφικές μηχανές τις δυνατότητες των μεγάλων; Όπως αντιλαμβάνεστε, οι διαφορές ανάμεσα στις μηχανές κάθε κατηγορίας πηγάζουν από το hardware και το firmware. Δυστυχώς, για το hardware δεν μπορούμε να κάνουμε πολλά. Θα ήταν από δύσκολο έως αδύνατο να αλλάξουμε τον αισθητήρα ή το φακό μιας compact, ενώ το κόστος θα προσέγγιζε εκείνο μιας dSLR. Μια τέτοια επέμβαση θα ήταν μάλλον ανούσια. Με το firmware, όμως, υπάρχει μεγάλη ευελιξία! Στο κάτω-κάτω, είναι πολύ ευκολότερο και σαφώς πιο

1. Επίσης, νιώθει πολύ παράξενα που γράφει για τον εαυτό του σε τρίτο πρόσωπο, αλλά θεωρεί ότι το να γράφει σε πρώτο είναι κάπως ψωνισμένο.
2. Ξέρουμε, ξέρουμε... Μόλις το (ξάνα)είπαμε, αλλά θέλαμε να καταλάβετε καλά τι είναι αυτό που δεν χρειάζεται να λέμε συχνά ;-)

ασφαλές να πειράξουμε το πρόγραμμα μιας μηχανής, σε σύγκριση με το εύθραυστο (κι ενίοτε ακριβό) hardware.

Προφανώς, για τις παραπάνω σκέψεις δεν θα κερδίζαμε κανένα βραβείο παγκόσμιας πρωτοτυπίας. Όλα αυτά τα έχουν σκεφτεί πολλοί άνθρωποι ανά τον κόσμο, ενώ υπάρχει και κάποιος που δεν έμεινε στη σκέψη και στη φαντασία. Μιλάμε για τον Andrey Gratchev, τον δημιουργό του Canon Hacking Development Kit! Το CHDK αποτελεί προϊόν ελεύθερου λογισμικού και είναι γραμμένο για τους επεξεργαστές DigIC (Digital Imaging Core), στους οποίους στηρίζονταν όλες οι compact της Canon, από το 2004 έως το 2011. Το εν λόγω πρόγραμμα προσδίδει στις μικρές φωτογραφικές μηχανές πολλά νέα χαρακτηριστικά, χωρίς να απομακρύνει ή να αντικαθιστά πλήρως το εργοστασιακό firmware. Εκτός από τον λεπτομερή έλεγχο της λειτουργίας της μηχανής, το CHDK επιτρέπει στους χρήστες να προσθέτουν δικά τους χαρακτηριστικά, όπως παιχνίδια και scripts σε UBASIC ή LUA. Ναι, καλά διαβάσατε: Θα μπορείτε να παίζετε στη φωτογραφική σας μηχανή, ενώ θα έχετε και τη δυνατότητα να αυτοματοποιείτε διάφορες εργασίες με τη βοήθεια των δικών σας script!


A410: 1.00e, 1.00f
A420: 1.00b
A430: 1.00b
A450: 1.00d
A460: 1.00d
A470: 1.00e, 1.01a, 1.01b, 1.02c
A480: 1.00b
A490: 1.00d, 1.00e, 1.00f
A495: 1.00d, 1.00e, 1.00f
A530: 1.00a
A540: 1.00b
A550: 1.00c
A560: 1.00a
A570: 1.00e, 1.01a
A580: 1.01b
A590: 1.00e, 1.01b
A610: 1.00d, 1.00e, 1.00f
A620: 1.00f, 1.00e
A630: 1.00c
A640: 1.00b
A650: 1.00d
A700: 1.00b
A710: 1.00a
A720: 1.00c
A800: 1.00b, 1.00c
A1000: 1.00a, 1.00b
A1100: 1.00c
A2000: 1.00c, 1.00a, 1.00b
A2100: 1.00a
A3000: 1.00b, 1.00c, 1.00d
A3200: 1.00d
A3300: 1.00a, 1.00c, 1.00d
D10: 1.00a, 1.00b
ELPH100 (IXUS 115): 1.00c, 1.01a, 1.01b,1.01c
ELPH300 (IXUS 220): 1.00c, 1.01a, 1.01c, 1.01d, 1.01e
ELPH310 (IXUS 230): 1.00b, 1.00e
ELPH500 (IXUS 310): 1.00a, 1.01a, 1.01b, 1.01c
S2: 1.00e, 1.00f, 1.00g, 1.00i
S3: 1.00a
S5: 1.01a, 1.01b
S90: 1.00c, 1.01a, 1.01b, 1.01c
S95: (beta) 1.00e, 1.00h, 1.00i, 1.00k
S100: 1.00d, 1.00e, 1.01a, 1.01b
G7: 1.00e, 1.00g, 1.00i, 1.00j
G9: 1.00d, 1.00f, 1.00g, 1.00h, 1.00i
G10: 1.02a, 1.03b, 1.04a
G11: (beta) 1.00f, 1.00j, 1.00k, 1.00l, 1.00h
G12: 1.00c, 1.00e, 1.00f, 1.00g
G1X: 1.00e, 1.00f, 1.00g
SD30 (IXUS izoom): 1.00g
SD300 (IXUS 40): 1.00j, 1.00k
SD400 (IXUS 50): 1.01a, 1.01b
SD450 (IXUS 55): 1.00b, 1.00c, 1.00d
SD500 (IXUS 700): 1.01a, 1.01b
SD550 (IXUS 750): 1.00f, 1.00g
SD600 (IXUS 60): 1.00a, 1.00d
SD630 (IXUS 65): 1.00a
SD700 (IXUS 800): 1.00b, 1.01b, 1.01a
SD750 (IXUS 75): 1.00b, 1.01a, 1.01b, 1.02a
SD770 (IXUS 85): 1.00a
SD780 (IXUS 100): 1.00b, 1.00c
SD790 (IXUS 90): 1.00c, 1.00d, 1.00e
SD800 (IXUS 850): 1.00d, 1.00e
SD850 (IXUS 950): 1.00c
SD870 (IXUS 860): 1.00c
SD880 (IXUS 870): 1.00e, 1.01a, 1.01b
SD890 (IXUS 970): 1.00b, 1.00c, 1.00d, 1.00f
SD900 (IXUS 900): 1.00c
SD940 (IXUS 120): 1.00e, 1.01a, 1.02c, 1.03b, 1.03c
SD950 (IXUS 960): 1.00c, 1.00d
SD980 (IXUS 200): 1.01c, 1.01d
SD990 (IXUS 980): 1.00e 1.01b
SD1000 (IXUS 70): 1.00c, 1.01a, 1.01b, 1.02a
SD1100 (IXUS 80): 1.00c, 1.01a, 1.01b
SD1200 (IXUS 95): 1.00c
SX1: 2.00h, 2.01a
SX10: 1.00c, 1.01a, 1.01b, 1.02b, 1.03a
SX20: 1.00f, 1.02b, 1.02c, 1.02d
SX30: 1.00e, 1.00h, 1.00l
SX40: (beta) 1.00d, 1.00f, 1.00g, 1.00i
SX100: 1.00b, 1.00c
SX110: 1.00b
SX120: 1.00b
SX130: 1.01c, 1.01d, 1.01f
SX150: 1.00a
SX200IS: 1.00c, 1.00d
SX210: 1.00c
SX220HS: 1.00a, 1.01a, 1.01b
SX230HS: 1.00c, 1.01a, 1.01b
TX1: 1.00g, 1.01b

Στην εικόνα φαίνονται όλα τα μοντέλα φωτογραφικών μηχανών που υποστηρίζει το CHDK. Μαζί με τα μοντέλα, εμφανίζεται και η έκδοση του firmware που απαιτείται για να λειτουργήσει το CHDK.

Σε αυτό το σημείο, πρέπει να τονίσουμε ότι το CHDK *δεν* καταστρέφει τη μηχανή. Ωστόσο, αν γράψουμε ένα script που ανοίγει το φακό ενώ το πορτάκι είναι κλειστό, ή αν μετακινήσουμε τους φακούς πέρα από τα κατασκευαστικά όρια, ενδέχεται να προκαλέσουμε μόνοι μας κάποια βλάβη... Δεν πιστεύουμε όμως να σας αποθάρρυνε αυτός ο κίνδυνος, ε; Εάν συνέβη κάτι τέτοιο, δείτε μια σύντομη λίστα των δυνατοτήτων του CHDK, για να γλυκαθείτε ξανά:

- Φωτογραφίες σε RAW, RAW Average και RAW Sum.
- Ιστόγραμμα σε πραγματικό χρόνο (RGB, blended κ.ά.)
- Εμφάνιση σε πραγματικό χρόνο των περιοχών με υπό και υπέρ έκθεση (Zebra mode).
- Υπολογισμός βάθους πεδίου, υπερεστίασης κ.ά.
- Ένδειξη επιπέδου φόρτισης της μπαταρίας, για μηχανές που δεν έχουν αυτό το χαρακτηριστικό.
- Μετρητές ελεύθερου χώρου για φωτογραφίες RAW κι ένδειξη διαθέσιμου χρονικού ορίου για το βίντεο.
- Απομακρυσμένος έλεγχος κλείστρου, μέσω USB.



Τοποθετώντας το κενό αρχείο ver.req στην κάρτα μνήμης και πατώντας ταυτόχρονα τα κουμπιά [FUNC/SET] και [DISPLAY], η μηχανή εμφανίζει διάφορες πληροφορίες και, μεταξύ άλλων, την έκδοση του εγκατεστημένου firmware.

- Αυτόματη ενεργοποίηση κάμερας σε κάθε ανίχνευση κίνησης. Σύστημα ικανό ακόμα και για την παγίδευση κεραυνών!
- Συνεχώς προσαρμοζόμενη ταχύτητα κλείστρου και διαφράγματος.
- Ρυθμιζόμενο bitrate και δέκα επίπεδα ποιότητας για τη συμπίεση των video.
- Εξάλειψη του περιορισμού του 1Gb για video recording (σε ορισμένες ψηφιακές).
- Zoom κατά τη λειτουργία λήψης video, για τις κάμερες που δεν έχουν αυτό το χαρακτηριστικό.
- Εξαιρετικά χαμηλές ταχύτητες κλείστρου, με χρόνο κλεισίματος πάνω από 64 δευτερόλεπτα.
- Υψηλές ταχύτητες κλείστρου, με χρόνο κλεισίματος της τάξης του 1/10000 sec.
- Υψηλές ταχύτητες συγχρονισμού φλας, της τάξης του 1/64000 sec.

Λήψη του CHDK

Κατ' αρχάς, πριν αρχίσουμε να ζητωκραυγάζουμε, πρέπει να τσεκάρουμε αν η μηχανή μας βρίσκεται στη λίστα με τις υποστηριζόμενες συσκευές. Η κοινότητα του CHDK ενημερώνει διαρκώς το λογισμικό και προστίθενται συνέχεια νέες μηχανές. Μπορούμε να μάθουμε αν ανήκουμε στους τυχερούς, εξετάζοντας τη δεξιά στήλη στην ακόλουθη σελίδα: http://chdk.wikia.com/wiki/CHDK. Εκεί εμφανίζεται μια λίστα με διάφορα μοντέλα της Canon, και δίπλα στο καθένα φαίνονται οι υποστηριζόμενες εκδόσεις του firmware. Βλέπετε, η εγκατάσταση του CHDK δεν προϋποθέτει μόνο συγκεκριμένες μηχανές, αλλά και συγκεκριμένες εκδόσεις του firmware! Πάντως, αν εντοπίσουμε το μοντέλο της μηχανής μας στη λίστα, αρκεί να κάνουμε κλικ επάνω του. Το ζήτημα με την έκδοση του firmware θα το αντιμετωπίσουμε λίγο παρακάτω. Εμείς πειραματιστήκαμε με μία Canon Powershot G9 και κάναμε κλικ στο "G9: 1.00d, 1.00f, 1.00g, 1.00h, 1.00i". Με αυτόν τον τρόπο μεταφερόμαστε σε μια νέα σελίδα, όπου περιγράφονται αναλυτικά οι εκδόσεις του firmware που υποστηρίζονται, τα όποια προβλήματα ενδέχεται να παρουσιαστούν, το πώς διορθώνονται και πάει λέγοντας. Εκεί, είναι φρόνιμο να διαβάσουμε (έστω και



Το firmware update πραγματοποιείται από τη σχετική επιλογή του κεντρικού μενού της μηχανής.

διαγώνια) όλα όσα έχουν να πουν οι προγραμματιστές του CHDK για τη μηχανή μας. Αμέσως μετά, μπορούμε να αναζητήσουμε και να κατεβάσουμε την κατάλληλη έκδοση του CHDK, η οποία προορίζεται για το μοντέλο και για το firmware της μηχανής μας. Όπως καταλαβαίνετε, αυτό προϋποθέτει να μάθουμε ποια έκδοση του firmware φοράει ήδη η μηχανή.



Για να προχωρήσει η εγκατάσταση του CHDK, αρκεί να πατήσουμε το κουμπί OK...

Αρχικά, βγάζουμε την κάρτα SD από τη μηχανή και τη συνδέουμε στον υπολογιστή, με κάποιον card reader. Στο root directory της κάρτας πρέπει να δημιουργήσουμε ένα κενό αρχείο, με το όνομα ver.req. Αυτό μπορεί να γίνει πανεύκολα, δημιουργώντας ένα κενό αρχείο text και μετονομάζοντάς το κατάλληλα. Βέβαια, αν δουλεύουμε σε Windows, για να γίνει σωστά αυτή η μετονομασία πρέπει να φροντίσουμε ώστε να εμφανίζονται οι επεκτάσεις στα ονόματα *όλων* των αρχείων. Κάτι τέτοιο γίνεται πολύ εύκολα στα windows 7. Αρκεί να ακολουθήσουμε την εξής διαδρομή: Πίνακας Ελέγχου → Εμφάνιση και Εξατομίκευση → Επιλογές φακέλων → Εμφάνιση κρυφών αρχείων και φακέλων. Εκεί θα πρέπει να πάμε στην καρτέλα "Προβολή" και να αφαιρέσουμε το τικ από την επιλογή "Απόκρυψη επεκτάσεων για γνωστούς τύπους αρχείων". Σε κάθε περίπτωση, αφού δημιουργήσουμε το ver.req, αρκεί να τοποθετήσουμε ξανά την κάρτα μνήμης στη φωτογραφική μηχανή. Κατόπιν την ανάβουμε σε play mode και πατάμε ταυτόχρονα τα κουμπιά [FUNC/SET] και [DISPLAY]. Σε μερικές κάμερες δεν υπάρχει το [DISPLAY], αλλά υπάρχει ένα ισοδύναμο κουμπί: το κάτω βελάκι. Σε αυτές τις κάμερες, ο παραπάνω συνδυασμός μετατρέπεται σε [FUNC/SET] και [↓]. Με αυτή τη διαδικασία, στην οθόνη της μηχανής θα εμφανιστεί η έκδοση του original firmware. Στη δική μας περίπτωση, μεταξύ άλλων πληροφοριών εμφανίστηκε και το εξής: Firmware Ver GM1.00H.



Το κεντρικό μενού της μηχανής, μετά την ενεργοποίηση του CHDK. Δύσκολα θα φανταστείτε κάποια ρύθμιση που θα λείπει από αυτό το μενού!

Γνωρίζοντας την έκδοση του firmware της μηχανής μας, μπορούμε να κατεβάσουμε τη σωστή εκδοχή του CHDK. Για το σκοπό αυτό αρκεί να επισκεφθούμε το

`http://mighty-hoernsche.de`

Εκεί εμφανίζεται ένας μεγάλος πίνακας με όλες τις υποστηριζόμενες συσκευές και με μια ξεχωριστή γραμμή για κάθε έκδοση του firmware. Σημειώστε ότι σε κάθε γραμμή προσφέρονται δύο link για κατέβασμα. Το πρώτο (το αριστερό) παραπέμπει στην πλήρη έκδοση του CHDK, ενώ το δεύτερο σε μια ελαφρώς περικομμένη. Εμείς πρέπει να επιλέξουμε την πλήρη (full). Έτσι, για τη δική μας φωτογραφική μηχανή κατεβάσαμε

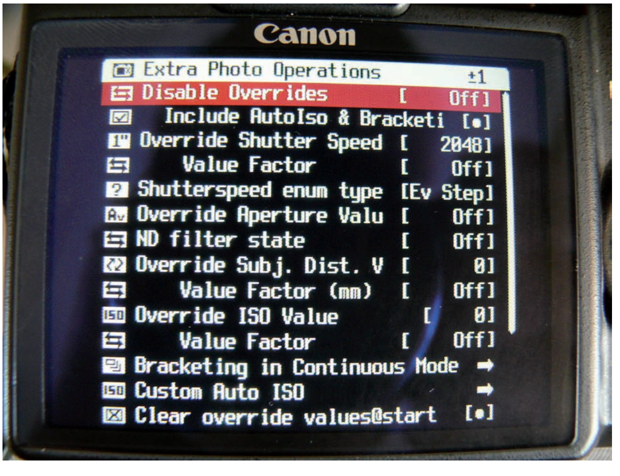


Στο υπομενού Miscellaneous υπάρχει ένας File Browser, ένα ημερολόγιο, ένας text reader, Παιχνίδια και διάφορα άλλα καλούδια!

το αρχείο g9-100h-1.1.0-2020-full.zip (709KB). Μετά κι από αυτό το βήμα, είμαστε έτοιμοι να προχωρήσουμε στην εγκατάσταση. Κάπου εδώ όμως πρέπει να διευκρινίσουμε ότι το CHDK δεν αντιγράφεται ποτέ σε κάποιο τσιπ της μηχανής. Επομένως, ο όρος "εγκατάσταση" είναι ελαφρώς παραπλανητικός. Στην ουσία, το CHDK φορτώνεται κι εκτελείται από τη μηχανή, κάθε φορά που την ενεργοποιούμε και *αφού* φορτωθεί το επίσημο firmware. Σκεφτείτε τη μηχανή σας σαν έναν υπολογιστή, το firmware σαν το λειτουργικό σύστημα, την κάρτα μνήμης σαν το σκληρό δίσκο και το CHDK σαν μια απλή εφαρμογή. Αυτή η εφαρμογή προϋποθέτει την εκκίνηση του λειτουργικού και μπορούμε να την ξεκινάμε όποτε θέλουμε εμείς -- ή να φροντίσουμε ώστε να ξεκινά αυτόματα. Στη συνέχεια θα περιγράψουμε τον χειροκίνητο αλλά και τον αυτόματο τρόπο εκκίνησης...

Χειροκίνητη έναρξη

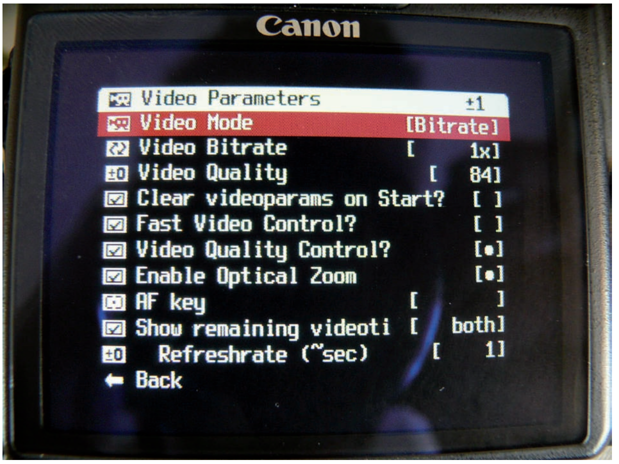
Αν σκοπεύουμε να χρησιμοποιούμε ελάχιστα το CHDK, δεν υπάρχει κανένας λόγος να το φορτώνουμε συνεχώς στη μνήμη της μηχανής. Σε αυτή την περίπτωση, λοιπόν, αρκεί να αποσυμπιέσουμε το πακέτο που κατεβάσαμε προηγουμένως και να αντιγράψουμε το περιεχόμενό του στο root directory της κάρτα SD. Στη συνέχεια, πρέπει να θέσουμε τη μηχανή σε playback mode και να την ενεργοποιήσουμε, κρατώντας πατημένο το κουμπί PLAY. Αν η μηχανή μας δεν έχει αυτό το κουμπί, τη θέτουμε στο σχετικό mode και την ανάβουμε κανονικά. Μετά, πατάμε το κουμπί MENU και αναζητάμε στο μενού της συσκευής την επιλογή Firm Update. Πατώντας το Firm Update, εμφανίζεται μια ερώτηση της μορφής "Update firmware version? 1.0.0.0 à 1.1.0.0". Αν απαντήσουμε θετικά (επιλέγοντας το OK), το CHDK θα φορτωθεί. Για να ενεργοποιηθεί, όμως, θα χρειαστεί να πατήσουμε και το κουμπί PRINT. Από εκείνη τη στιγμή και μέχρι να σβήσουμε τη μηχανή, το λογισμικό της θα είναι εμπλουτισμένο με τις δυνατότητες του CHDK. Για να δούμε το αναβαθμισμένο μενού της συσκευής, αρκεί να πατήσουμε το κουμπί MENU. Πάντως, αν σβήσουμε τη μηχανή, την επόμενη φορά που θα την ενεργοποιήσουμε, το μενού του CHDK θα έχει εξαφανιστεί. Βλέπετε, το firmware update που πραγματοποιείται με τον παραπάνω τρόπο, επιδρά μόνο στην προσωρινή μνήμη του συστήματος και δεν είναι μόνιμο.



Η εικόνα μιλάει από μόνη της: Extra Photo Operations. Πείτε μας, δεν θέλετε να ξεκινήσετε αμέσως τα πειράματα;

Αυτόματη εκτέλεση

Αν η χειροκίνητη εκκίνηση του CHDK μοιάζει στα μάτια μας με κακό μπελά, μπορούμε να φροντίσουμε ώστε να εκτελείται αυτόματα. Όμως, για να πετύχουμε κάτι τέτοιο, πρέπει να προετοιμάσουμε κατάλληλα την κάρτα μνήμης κι αυτό δεν σημαίνει μια απλή αντιγραφή. Για να λειτουργήσει η αυτόματη εκκίνηση του CHDK, η κάρτα μας θα πρέπει να έχει συγκεκριμένα χαρακτηριστικά, τα οποία εξαρτώνται από τη χρονολογία κατασκευής της μηχανής μας. Μπορείτε να βρείτε όλες τις σχετικές λεπτομέρειες εδώ:



Video Parameters: Από εδώ ρυθμίζουμε την ποιότητα του video, το Bitrate, το αν θα μπορεί να γίνει οπτικό zoom κατά τη διάρκεια της εγγραφής κ.ά.

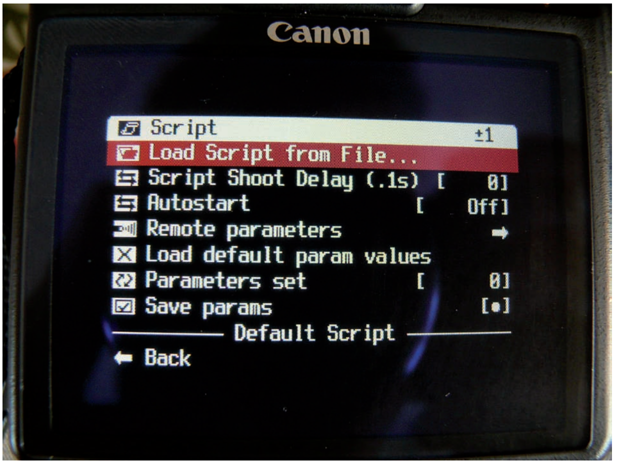
```
http://chdk.wikia.com/wiki/Prepare_your_SD_card
```

Πάντως, στις περισσότερες περιπτώσεις, μια κάρτα των 4GB με μία μόνο κατάτμηση και φορμαρισμένη σε FAT16, θα παίξει κανονικά! Στη συνέχεια, λοιπόν, θα δούμε την προετοιμασία μιας τέτοιας κάρτας και την ενεργοποίηση της αυτόματης εκκίνησης.

Το πιο κρίσιμο βήμα της διαδικασίας αφορά στο σωστό format της κάρτας. Αν δουλεύουμε σε Windows, αρκεί να μάθουμε ποιο γράμμα αντιστοιχεί στην κάρτα μνήμης και να ανοίξουμε τη γραμμή εντολών. Εκεί, αν υποθέσουμε ότι στην κάρτα μας αντιστοιχεί το γράμμα F, αρκεί να δώσουμε κάτι τέτοιο:

```
C:\Users\Drg> format F: /
fs:fat
```

Προφανώς, ο καθένας θα πρέπει να βάλει στη θέση του F το γράμμα που αντιστοιχεί στην κάρτα του. Αυτό το τονίζουμε (παρά τον κίνδυνο να γίνουμε κουραστικοί), διότι αν δεν φανούμε προσεκτικοί ενδέχεται να χάσουμε πολύτιμα δεδομένα ή/και να καταστρέψουμε την εγκατάσταση του λειτουργικού. Για το σωστό φορμάρισμα της κάρτας, οι χρήστες του Linux θα πρέπει επίσης να στραφούν στη γραμμή εντολών. Για παράδειγμα, όσοι χρησιμοποιούν το Ubuntu ή κάποια συγγενική διανομή, μπορούν να δώσουν κάτι σαν κι αυτό:



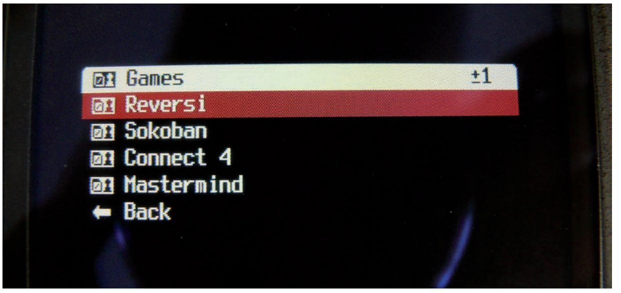
Το περιβόητο μενού Script: Από εδώ φορτώνουμε τα δικά μας προγράμματα στη φωτογραφική μηχανή!

```
# sudo mkfs.msdos -F 16 /dev/sdd
```

Στην περίπτωση που το mkfs.msdos απουσιάζει από το σύστημά μας, μπορούμε να το προσθέσουμε δίνοντας αυτό:

```
# sudo apt-get install dosfstools
```

Προφανώς, όπως και στην περίπτωση των Windows, κατά την εκτέλεση του format οφείλουμε να είμαστε ιδιαίτερα προσεκτικοί! Στη θέση του



Τίποτα δεν λείπει από το CHDK. Ούτε καν τα παιχνίδια! Η πλήρης έκδοση του προγράμματος ενσωματώνει τα Reversi, Sokoban, Mastermind και Σκορ 4...

/dev/sdd θα πρέπει να βάλει ο καθένας το device file που έχει αντιστοιχηθεί στην κάρτα του...

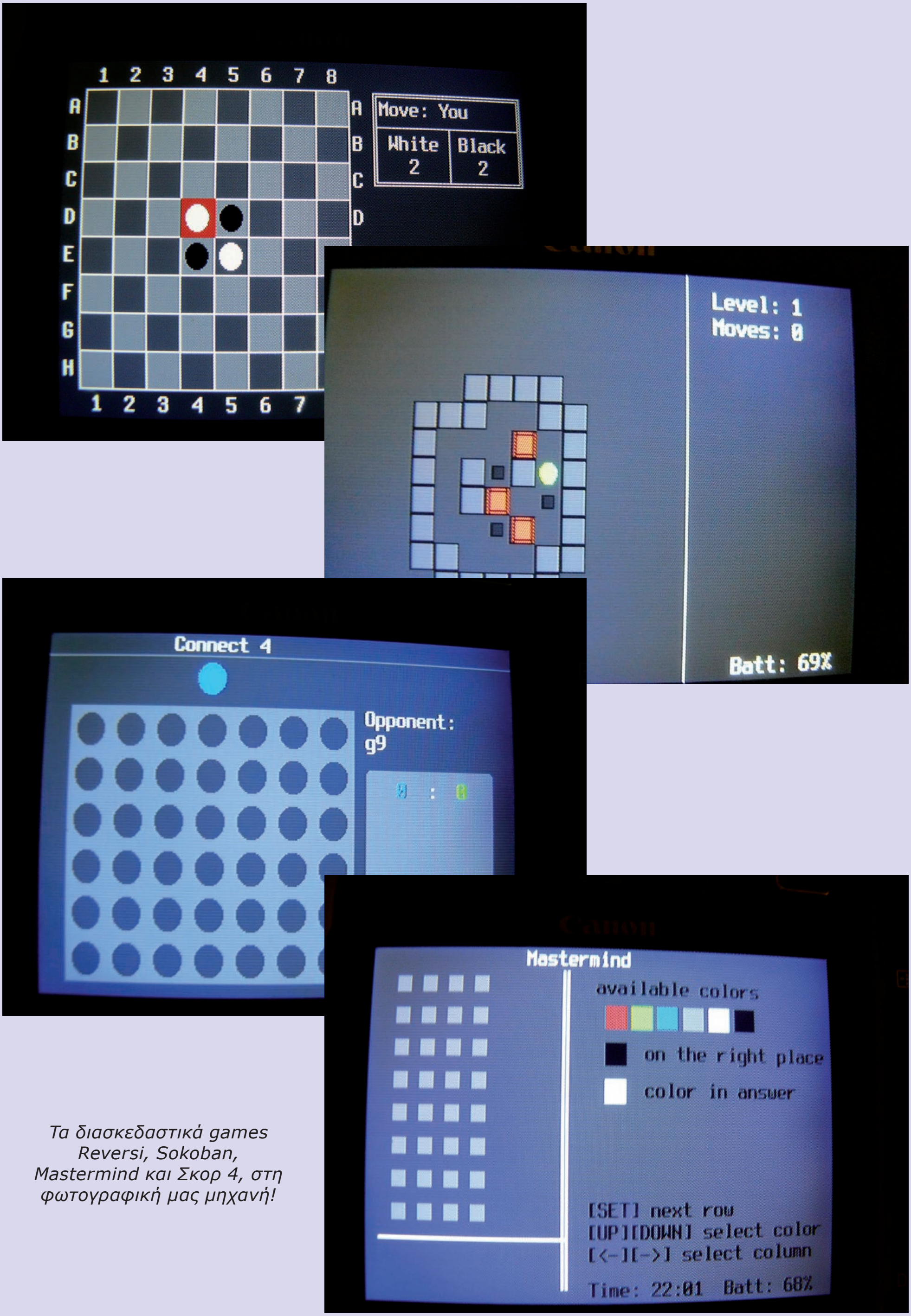
Αφού ολοκληρωθεί η διαμόρφωση, αποσυμπιέζουμε το πακέτο με το CHDK και τοποθετούμε το περιεχόμενό του στην κάρτα -- χύμα! Κατόπιν, τοποθετούμε την κάρτα στην ψηφιακή μηχανή και θέτουμε την τελευταία σε playback mode. Στη συνέχεια την ενεργοποιούμε, κρατώντας πατημένο το κουμπί PLAY. Σε αυτό το σημείο ισχύουν αυτά που είπαμε και προηγουμένως. Μετά, από το κεντρικό μενού της συσκευής, επιλέγουμε το firmware update. Όταν ολοκληρωθεί η φόρτωση του CHDK πατάμε το κουμπί PRINT για να το ενεργοποιήσουμε. Για να εξασφαλίσουμε την αυτόματη φόρτωση του CHDK, επιλέγουμε από το κεντρικό μενού την επιλογή “Miscellaneous Stuff”. Έτσι θα εμφανιστεί ένα νέο μενού, στο οποίο θα πρέπει να επιλέξουμε το “Make Card Bootable”. Μόλις ολοκληρωθεί η διαδικασία, κλείνουμε την φωτογραφική μηχανή, βγάζουμε την κάρτα SD και την κλειδώνουμε! Αυτό γίνεται με το μικρό, συρόμενο διακοπτάκι που διαθέτουν στο πλάι τους οι κάρτες SD. Αυτό ήταν όλο! Από εδώ και πέρα, κάθε φορά που θα ανοίγουμεε τη φωτογραφική μηχανή θα μπορούμε να παραμένουμε στο εργοστασιακό firmware ή να πατάμε το πλήκτρο PRINT και να ενεργοποιούμε ακαριαία το CHDK. Παρεμπιπτόντως, να πούμε ότι το κλείδωμα της κάρτας δεν πρέπει να μας ανησυχεί καθόλου. Πρόκειται για έναν μηχανισμό ασφάλειας στο software και όχι στο hardware, τον οποίο παρακάμπτει αυτόματα το CHDK. Επομένως, θα μπορούμε να τραβάμε φωτογραφίες και βίντεο χωρίς το παραμικρό πρόβλημα, ακόμα και με την κάρτα κλειδωμένη.

Επίλογος

Το CHDK είναι ένα δυνατό εργαλείο στα χέρια του έμπειρου αλλά και του ερασιτέχνη φωτογράφου. Μπορούμε να σας πούμε με σιγουριά ότι το CHDK προσφέρει όλες τις δυνατότητες που θα θέλατε να έχει το software της μηχανής σας και σίγουρα περισσότερες από όσες φανταζόσασταν. Εξάλλου, η δυνατότητα εκτέλεσης custom script (σε LUA ή σε UBASIC), επεκτείνει απεριόριστα το πεδίο εφαρμογών της μηχανής σας. Σε επόμενο τεύχος θα επανέλθουμε στο θέμα και θα παρουσιάσουμε τα δικά μας scripts. Μέχρι τότε, σας προτείνουμε να παίξετε με το CHDK και να μελετήσετε το επίσημο tutorial:

http://chdk.wikia.com/wiki/File:CHDK_Quick_Start_User_Guide_-_Booklet.pdf

Επιπρόσθετα, αν έχετε ήδη σκεφτεί μερικές ωραίες ιδέες για script, μπορείτε να ασχοληθείτε και με το scripting. Αν θέλετε να μάθετε τη LUA, ξεκινήστε από εδώ: http://chdk.wikia.com/wiki/Lua. Διαφορετικά, αν θέλετε να ασχοληθείτε με την UBASIC, κοιτάξτε εδώ http://chdk.wikia.com/wiki/UBASIC κι εδώ http://chdk.wikia.com/wiki/UBASIC/TutorialScratchpad. Τέλος, περισσότερα για το scripting, καθώς και διάφορες ιδέες που θα τροφοδοτήσουν την φαντασία σας, μπορείτε να βρείτε στο forum του CHDK: http://chdk.setepontos.com. Για την ώρα, σας αφήνουμε να παίξετε με τον νέο σας υπολογ... εχμ, με την αναβαθμισμένη σας φωτογραφική μηχανή, θέλαμε να πούμε :)



Τα διασκεδαστικά games Reversi, Sokoban, Mastermind και Σκορ 4, στη φωτογραφική μας μηχανή!



Skill: Beginner

Tags: IRC, InspIRCd, user modes, cloaking, hashed passwords

(Το προσωπικό μας)

Διαχρονικό στέκι επικοινωνίας

Οι σύγχρονες υπηρεσίες επικοινωνίας μεταφέρουν κείμενο, φωνή και εικόνα, προσφέροντας ταυτόχρονα υψηλά επίπεδα ασφάλειας. Ωστόσο, για καμία δεν είμαστε σίγουροι ότι θα συνεχίσει να υπάρχει μετά από δέκα χρόνια. Αντίθετα, το εξαιρετικά λιτό IRC, που δεν προσφέρει καμία φανταχτερή λειτουργία παρά μόνο συνομιλία σε πραγματικό χρόνο μέσω απλού κειμένου, έχει κλείσει πάνω από 20 χρόνια ζωής! Τι θα λέγατε να στήναμε τον δικό μας IRC server; Φανταστείτε τον σαν ένα ιδιαίτερο μέρος, στο οποίο θα συναντιόμαστε με τα φιλαράκια μας...

by giannoug

Το mIRC είναι ο διασημότερος client για IRC δίκτυα κι ευρέως διαδεδομένος στο GRNet. Ο προγραμματισμός του ξεκίνησε το 1994, αλλά η εξέλιξή του συνεχίζεται μέχρι σήμερα και δέχεται συχνές αναβαθμίσεις!



Είναι ξανά 1996. Το Internet στην Ελλάδα κάνει τα πρώτα του δειλά βήματα στα σπίτια. Μαζί του έρχεται και το online chat, με τους πρώτους ελληνικούς IRC servers. Ένα χρόνο μετά θα δημιουργηθεί το GRnet και μέσα σε λίγους μήνες θα αριθμεί 20 servers. Το Σεπτέμβρη της ίδιας χρονιάς θα εμφανιστεί και η nana: Ένας από τους δημοφιλέστερους IRC servers μέχρι σήμερα! Η nana συμπληρώνει πλέον δεκαέξι χρόνια λειτουργίας, έχοντας εξυπηρετήσει χιλιάδες χρήστες και άπειρες, ατελείωτες συζητήσεις... Όλα αυτά, χάρη στο πρωτόκολλο IRC (Internet Relay Chat). Τι θα λέγατε αν φτιάχναμε κάτι παρόμοιο; Φυσικά, δεν θα κάτσουμε να στήσουμε δεκάδες server, ούτε φιλοδοξούμε να φιλοξενήσουμε χιλιάδες χρήστες ταυτόχρονα. Θα σηκώσουμε έναν απλό IRC server, για καθαρά "εσωτερική" χρήση. Ο server μας θα υποστηρίζει όλες τις σύγχρονες λειτουργίες του πρωτοκόλλου IRC και φυσικά θα είναι λίγο πιο ασφαλής από τους αντίστοιχους δημόσιους. Όπως καταλαβαίνετε, γι' αυτή τη δουλειά θα στραφούμε στο Linux. Εμείς εργαστήκαμε σ' ένα σύστημα με το Ubuntu 12.04[1], αλλά τα βήματα που θα δούμε θα μπορούσαν να μεταφερθούν αυτούσια σε οποιαδήποτε άλλη διανομή. Εξαίρεση αποτελεί η εγκατάσταση του σχετικού daemon, αλλά είμαστε σίγουροι ότι δεν θα σας σταματήσει ένα τόσο μικρό εμπόδιο :)

InspIRCd IRC daemon

Υπάρχουν δεκάδες IRC server daemons, καθένας από τους οποίους έχει τα θετικά και τα αρνητικά του. Εμείς επιλέξαμε τον InspIRCd (`http://www.inspircd.org`),

1. Περιττό να σημειώσουμε πως αν το Ubuntu τρέχει σε κάποιο VPS, τότε ακόμα καλύτερα!

GRNet Servers

Οι παρακάτω είναι οι servers που αποτελούν σήμερα το GRNet

1. aetos.irc.gr, TEI Θεσσαλονίκης, admin: Boldtrick, SSL port +6970
2. darkness.irc.gr, OVH, Παρίσι, admin: Dinos, SSL port +9999
3. dionisos.irc.gr, Hellas OnLine, Αθήνα, admin: george123
4. global.irc.gr, Telnet, Βουλγαρία, admin: HSOC, SSL port +7010
5. jersey, RackVibe στο New Jersey, USA, admin: MaGoS, SSL port +9667
6. nana.irc.gr, Forthnet, Θεσσαλονίκη, admin: SotirakiS, SSL port +9999
7. nemesis.irc.gr, CYTANET, Κύπρος, admin: WiZy, SSL port +7010
8. otrere.irc.gr, Πανεπιστήμιο Κρήτης, Ηράκλειο, admin: BILIAS, SSL port +9667
9. rea.irc.gr, Πανεπιστήμιο Κρήτης, Ηράκλειο, admin:Greekman, SSL port +9667
10. phobos.irc.gr, VServers, Γερμανία, admin: WiZy, SSL port +7010
11. tiger.irc.gr, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, admin: john, SSL port +7010
12. uk.irc.gr, RapidSwitch, UK, admin: Jester, SSL port +7443
13. virginia.irc.gr, Comcast, USA, admin: Yanni, SSL port +7010
14. warrior.irc.gr, LeaseWeb Datacenter, Netherlands, admin: Jester, SSL port +7443

καθώς υπερκαλύπτει άνετα τις ανάγκες μας, είναι εύκολος στη χρήση και υποστηρίζει όλες τις σύγχρονες λειτουργίες του πρωτοκόλλου IRC. Η εγκατάστασή του στο Ubuntu πραγματοποιείται πανεύκολα με τον συνηθισμένο τρόπο:

```
sudo apt-get install inspircd
```

Τα αρχεία ρυθμίσεων του IRC daemon βρίσκονται στον κατάλογο /etc/inspircd. Πριν ωστόσο καταπιαστούμε μαζί τους θα χρειαστεί να κάνουμε μια μικρή επέμβαση, σε κάποιο άλλο αρχείο του server. Συγκεκριμένα, θα πρέπει να ανοίξουμε το /etc/default/inspircd:

```
sudo nano[2] /etc/default/inspircd
```

Σ’ αυτό το αρχείο θα αλλάξουμε την τιμή του INSPIRCD_ENABLED από 0 σε 1. Όπως αντιλαμβάνεστε, με αυτή την αλλαγή ενεργοποιούμε τη λειτουργία του server. Αναρωτιέστε γιατί δεν ενεργοποιείται αυτόματα, αμέσως μετά την εγκατάσταση; Η μικρή επέμβαση που είδαμε αποτελεί μια δικλείδα ασφαλείας, για τους απρόσεκτους/βιαστικούς χρήστες οι οποίοι εγκαθιστούν υπηρεσίες και τις εκτελούν με τις “εργοστασιακές” ρυθμίσεις. Βλέπετε, οι προεπιλεγμένες ρυθμίσεις του InspIRCD έχουν πολλά κενά ασφάλειας και η επέμβαση που μόλις περιγράψαμε αποτελεί μια υπενθύμιση προς το χρήστη, ότι θα πρέπει να τσεκάρει και να προσαρμόσει κατάλληλα τις ρυθμίσεις! Αφού κάνουμε την αλλαγή, αποθηκεύουμε το αρχείο [CTRL+O] και κλείνουμε το nano [CTRL+X]. Τώρα ο InspIRCd μπορεί να ξεκινήσει, αλλά πριν από αυτό θα τον φέρουμε στα μέτρα μας. Ο φάκελος των ρυθμίσεων (/etc/inspircd/) περιέχει τρία μόλις αρχεία: Το inspircd.conf, το motd και το rules. Το πρώτο αποτελεί το κύριο αρχείο ρυθμίσεων. Το δεύτερο και το τρίτο περιλαμβάνουν το “μήνυμα της

Το σημερινό δίκτυο GRNet IRC είναι ελάχιστα μικρότερο από ό,τι ήταν στην αρχή του. Το hardware του είναι σαφώς αναβαθμισμένο και οι χρήστες του, αν και λιγότεροι, διατηρούν την ίδια αγάπη για το IRC.

2. Το nano αποτελεί έναν απλό και σχετικά εύχρηστο editor, για αρχεία απλού κειμένου. Αν δεν είναι εγκατεστημένο στο σύστημά μας μπορούμε να το προσθέσουμε εκτελώντας το “sudo apt-get install nano”.

ημέρας" (Message Of The Day) και τους κανόνες ορθής λειτουργίας του server. Ουσιαστικά, πρόκειται για αρχεία απλού κειμένου, το περιεχόμενο των οποίων εμφανίζεται αυτόματα κατά τη σύνδεση κάθε χρήστη. Εμείς θα εστιάσουμε στο κύριο αρχείο ρυθμίσεων, το οποίο και θα ανοίξουμε με το nano:

```
sudo nano /etc/inspircd/inspircd.conf
```

Όπως (σχεδόν) κάθε αρχείο ρυθμίσεων, έτσι και το inspircd.conf είναι χωρισμένο σε τμήματα. Καθένα από αυτά καθορίζει τη συμπεριφορά μιας επιμέρους λειτουργίας του server. Εμείς θα ασχοληθούμε με αρκετές ενότητες, ξεκινώντας από την αρχή...

```
<server name="irc.my-server.gr"
        description=" GianouG's awesome IRC server."
        network="GG IRC network">

<admin name="Giannou Gorge"
       nick="giannoug"
       email="gg_irc@gmail.com">
```

Στις πρώτες γραμμές συναντάμε μερικές βασικές παραμέτρους, για τον νέο μας IRC server. Πρώτα απ' όλα, ορίζουμε το hostname. Το όνομα του server, δηλαδή, το οποίο θα χρησιμοποιεί κανείς για να συνδεθεί. Στη συνέχεια δίνουμε μια μικρή περιγραφή του server και την επιθυμητή ονομασία για το δίκτυο στο οποίο ανήκει. Όπως αντιλαμβάνεστε, αυτά τα δύο μεγέθη έχουν αυθαίρετο περιεχόμενο. Ωστόσο, αν θελήσουμε στο μέλλον να προσθέσουμε έναν ακόμα sever, για "παράλληλη" λειτουργία με τον ήδη υπάρχοντα, θα πρέπει και οι δυο τους να ανήκουν στο ίδιο δίκτυο. Στη συνέχεια πρέπει να δώσουμε κάποια στοιχεία για εμάς. Ένα όνομα, ένα nickname και το email μας αρκούν, για οποιονδήποτε θέλει να επικοινωνήσει μαζί μας.

Λίγο χαμηλότερα βρίσκεται ο ορισμός του λεγόμενου bind address. Με απλά λόγια, πρόκειται για τη διεύθυνση στην οποία θα "ακούει" ο server μας. Στην ίδια γραμμή δηλώνουμε και τη δικτυακή θύρα (port) στην οποία θα δέχεται συνδέσεις. Εμείς τροποποιήσαμε την εν λόγω γραμμή ως εξής:

```
<bind address="" port="6667" type="clients">
```

Με το να *μην* ορίσουμε μια συγκεκριμένη διεύθυνση, ο IRC daemon θα ακούει σε όλες τις διευθύνσεις, όλων των καρτών δικτύου του μηχανήματός. Οι συνδέσεις που θα δέχεται, όμως, θα πρέπει να κατευθύνονται στο TCP port 6667. Θα μπορούσαμε να επιλέξουμε οποιοδήποτε άλλο port, αλλά το 6667 είναι το "επίσημο" port για το IRC και είναι προεπιλεγμένο σε όλους τους IRC clients.

Σε πολύ κρίσιμες καταστάσεις, ο InspIRCd επιτρέπει στους διαχειριστές καναλιών (operators) να διακόπτουν ή να επανεκκινούν τη λειτουργία του *χωρίς* να έχουν δικαιώματα root στο σύστημα. Φυσικά, για κάνει κάτι τέτοιο ένας operator, πρέπει να γνωρίζει ορισμένους κωδικούς. Αυτούς μπορούμε να τους δηλώσουμε λίγο χαμηλότερα στο αρχείο ρυθμίσεων, από την ακόλουθη γραμμή:

```
<power diepass="polu_mustikos_kwdikos" restartpass="arketa_
mustikos_kwdikos">
```

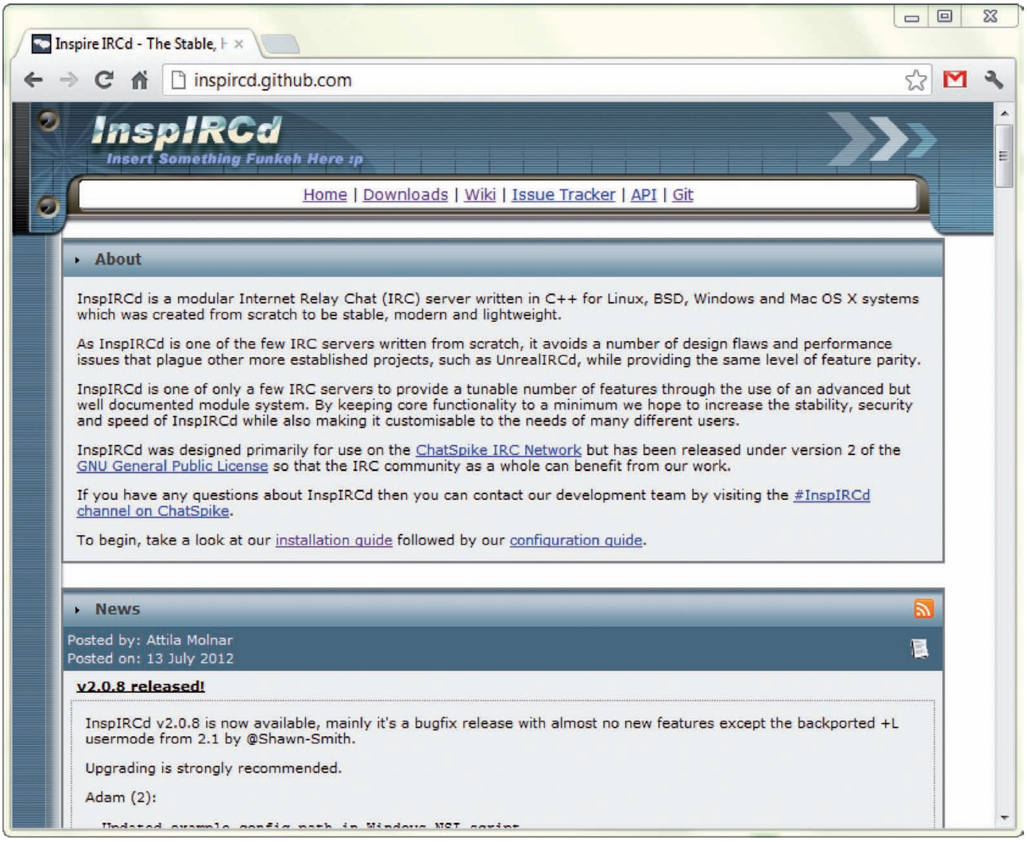
Περιττό να σημειώσουμε ότι αυτοί οι κωδικοί πρέπει να είναι πολύ δύσκολοι/σύνθετοι, αλλά κι εμείς πρέπει να προσέχουμε ιδιαίτερα πότε και σε ποιους τους γνωστοποιούμε. Όπως καταλαβαίνετε, αυτοί οι κωδικοί

αποτελούν την ύστατη λύση (σε μια κατάσταση την οποία αδυνατούμε να φανταστούμε ;).

Το σύστημα διαχείρισης δικαιωμάτων στο InspIRCd είναι αρκετά απλό και ταυτόχρονα πολύ αποτελεσματικό. Σε πρώτη φάση, οι διάφορες εντολές του IRC χωρίζονται σε ομάδες, οι οποίες ονομάζονται κλάσεις. Παρακάτω φαίνονται οι κλάσεις εντολών που περιλαμβάνει εξ ορισμού το αρχείο ρυθμίσεων:

```
<class name="Shutdown" commands="DIE RESTART REHASH LOADMODULE
UNLOADMODULE RELOAD">
<class name="ServerLink" commands="CONNECT SQUIT RCONNECT MKPASSWD
MKSHA256">
<class name="BanControl" commands="KILL GLINE KLINE ZLINE QLINE
ELINE">
<class name="OperChat" commands="WALLOPS GLOBOPS SETIDLE SPYLIST
SPYNAMES">
<class name="HostCloak" commands="SETHOST SETIDENT SETNAME CHGHOST
CHGIDENT">
```

Σε δεύτερη φάση, ορίζονται οι κατηγορίες χρηστών. Σε κάθε κατηγορία αντιστοιχίζονται διάφορες ομάδες εντολών και με αυτόν τον τρόπο προσδιορίζεται με ακρίβεια το τι επιτρέπεται να κάνει ο χρήστης κάθε κατηγορίας. Δείτε τις σχετικές γραμμές από το αρχείο ρυθμίσεων:



Η επίσημη σελίδα του InspIRCd θυμίζει έντονα τη δεκαετία του 90. Την εποχή, δηλαδή, κατά την οποία το IRC μεσουρανούσε.

```
<type name="NetAdmin" classes="OperChat BanControl HostCloak
Shutdown ServerLink" host="netadmin.irc.example.gr">
<type name="GlobalOp" classes="OperChat BanControl HostCloak
ServerLink" host="ircop.irc.example.gr">
<type name="Helper" classes="HostCloak" host="helper.irc.example.
gr">
```

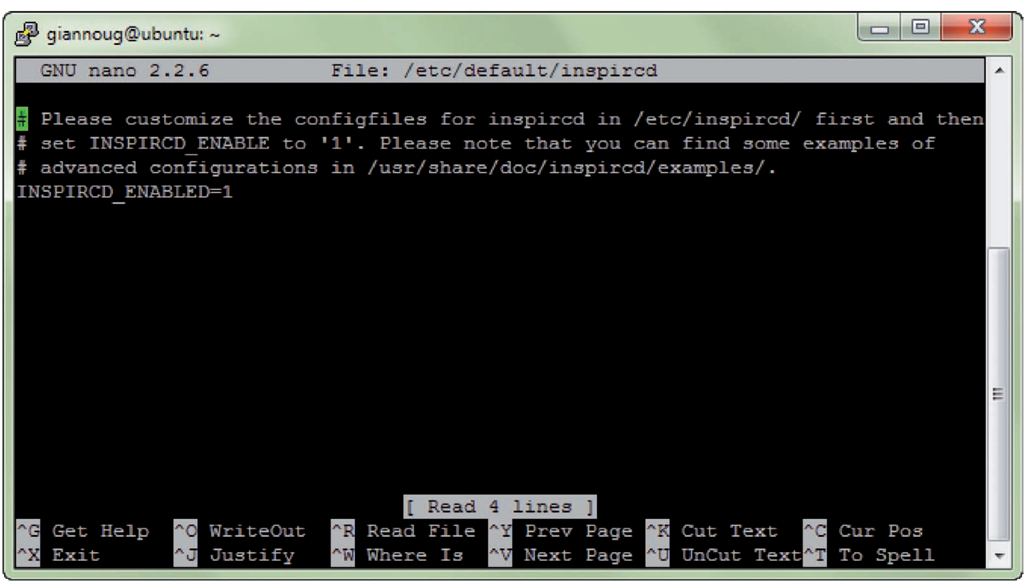
Με βάση τις συγκεκριμένες δηλώσεις, ένας χρήστης Global IRC operator (GlobalOp) *δεν* επιτρέπεται να ανοιγοκλείνει το server ή να προσθαφαιρεί modules, εφόσον *δεν* έχει πρόσβαση στις εντολές της κλάσης Shutdown. Φυσικά, εμείς μπορούμε να τροποποιήσουμε τις κλάσεις εντολών ή/και να ορίσουμε νέες. Παρόμοιες ελευθερίες έχουμε και με τις κατηγορίες χρηστών. Νομίζουμε ότι η απλότητα αλλά και η τεράστια ευελιξία που προσφέρει αυτό το σύστημα διαχείρισης δικαιωμάτων είναι πλέον εμφανής. Ορίζοντας τις κατάλληλες κλάσεις εντολών και τις κατάλληλες ομάδες χρηστών, μπορούμε να καθορίσουμε με σαφήνεια και με ευκολία το τι επιτρέπεται (και κατ' επέκταση το τι *δεν* επιτρέπεται) να κάνουν οι διάφοροι χρήστες.

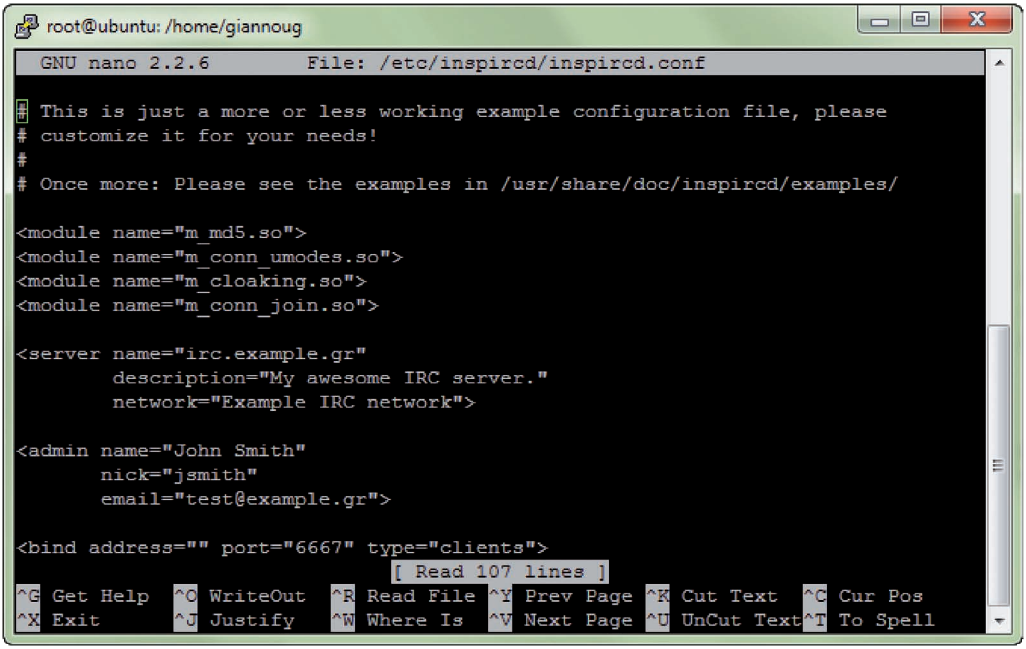
Μετά από αυτές τις γενικές ρυθμίσεις, ακολουθούν κάποιες πιο ειδικές! Συγκεκριμένα, στις επόμενες γραμμές ορίζεται ο πρώτος και ισχυρότερος χρήστης του IRC server. Δείτε τις σχετικές γραμμές από το αρχείο ρυθμίσεων, όπως τις τροποποιήσαμε:

```
<oper name="giannoug"
      password="heavy_melon_password"
      host="*"
      type="NetAdmin">
```

Η πρώτη επέμβαση που πρέπει να κάνουμε προκειμένου να πείσουμε το inspIRCd ότι ασχοληθήκαμε με τη ρύθμισή του και δεν είμαστε απρόσεκτοι, είναι στο αρχείο /etc/default/inspircd.

Το μόνο που χρειάζεται να ορίσουμε εδώ είναι ένα username, ένα password και το host address, από το οποίο θα συνδεόμαστε. Όπως βλέπετε, εμείς δεν ήμασταν και πολύ σοβαροί όταν επιλέγαμε password. Εσείς όμως καλά θα κάνετε να είσαστε ιδιαίτερα προσεκτικοί και να επιλέξετε ένα ισχυρό συνθηματικό. Στο πεδίο host δώσαμε σαν τιμή τον αστερίσκο, ώστε η σύνδεσή μας να επιτρέπεται από οπουδήποτε! Αν εσείς





Όλες οι ρυθμίσεις του IRC server μας βρίσκονται στο αρχείο inspircd.conf.

σκοπεύετε να συνδέεστε από κάποιο συγκεκριμένο μηχάνημα, μπορείτε να κάνετε τη λειτουργία του server ασφαλέστερη δίνοντας στο πεδίο host τη διεύθυνση του μηχανήματός σας.

Θυμόσαστε τα αρχεία motd και rules, τα οποία συγκατοικούν με το αρχείο ρυθμίσεων, εντός του καταλόγου /etc/inspircd; Ε, λοιπόν, το περιεχόμενο των συγκεκριμένων αρχείων δεν εμφανίζεται υποχρεωτικά σε κάθε χρήστη που συνδέεται. Από το αρχείο ρυθμίσεων μπορούμε να ορίσουμε οποιοδήποτε αρχείο επιθυμούμε να εμφανίζεται στη θέση του motd ή των κανόνων λειτουργίας. Αυτό γίνεται από τις ακόλουθες γραμμές:

```
<files motd="motd"
       rules="rules">
```

Ουσιαστικά, με την παραπάνω μικροεπέμβαση ολοκληρώνεται η βασική ρύθμιση του inspIRCd. Για να τον ξεκινήσουμε, αρκεί να δώσουμε κάτι τέτοιο:

```
sudo service inspircd restart
```

Πλέον, ο server θα είναι έτοιμος και λειτουργικός, θα δέχεται συνδέσεις και θα μπορεί να φιλοξενεί τις συζητήσεις μας. Μπορούμε να συνδεθούμε σε αυτόν με οποιονδήποτε IRC client θέλουμε. Πάντως, για τα Windows θα σας προτείνουμε το mIRC και για το Linux το XChat. Έχετε υπόψη ότι αφού συνδεθείτε στον IRC server σας μπορείτε να "αναβαθμιστείτε" σε διαχειριστές, εκτελώντας την παρακάτω εντολή από τον client:

```
/OPER giannoug heavy_melon_password
```

Φυσικά, στη θέση των giannoug και heavy_melon_password θα πρέπει να δώσετε το username και το αντίστοιχο password, που δηλώσατε στο αρχείο ρυθμίσεων.

Αυτόματη σύνδεση σε κανάλια

Ο δικός μας server θα είναι κατά πάσα πιθανότητα μικρός και θα χρησιμοποιείται από περιορισμένο αριθμό χρηστών. Μάλιστα, βέβαιο είναι ότι οι χρήστες του λίγο-πολύ θα γνωρίζονται μεταξύ τους. Επομένως, θα ήταν ιδιαίτερα βολικό αν συνδέονταν όλοι αυτόματα σε συγκεκριμένα κανάλια του server. Για μια στιγμή! Σε συγκεκριμένα κανάλια; Ένας τόσο μικρός server θα μπορούσε να έχει μόνο ένα κανάλι. Εμείς, τουλάχιστον, όταν αποφασίσαμε να φτιάξουμε τον δικό μας server, φανταζόμασταν όλο τον κόσμο συγκεντρωμένο σε ένα κανάλι, για να γίνεται και περισσότερος χαβαλές. Ακριβώς γι’ αυτό, θα μας άρεσε πάρα πολύ αν κάθε χρήστης συνδεόταν αυτόματα σε κάποιο κανάλι. Ε, λοιπόν, κάτι τέτοιο μπορεί να γίνει πανεύκολα με τον inspIRCd. Ο IRC server που επιλέξαμε διαθέτει πάρα πολλά modules, τα οποία επεκτείνουν τη λειτουργικότητά του. Ένα από αυτά, το m_conn_join, πετυχαίνει ακριβώς αυτό που περιγράψαμε παραπάνω. Για να το ενεργοποιήσουμε, αρκεί να ανοίξουμε το αρχείο ρυθμίσεων και να προσθέσουμε την ακόλουθη γραμμή:

```
<module name="m_conn_join.so">
```

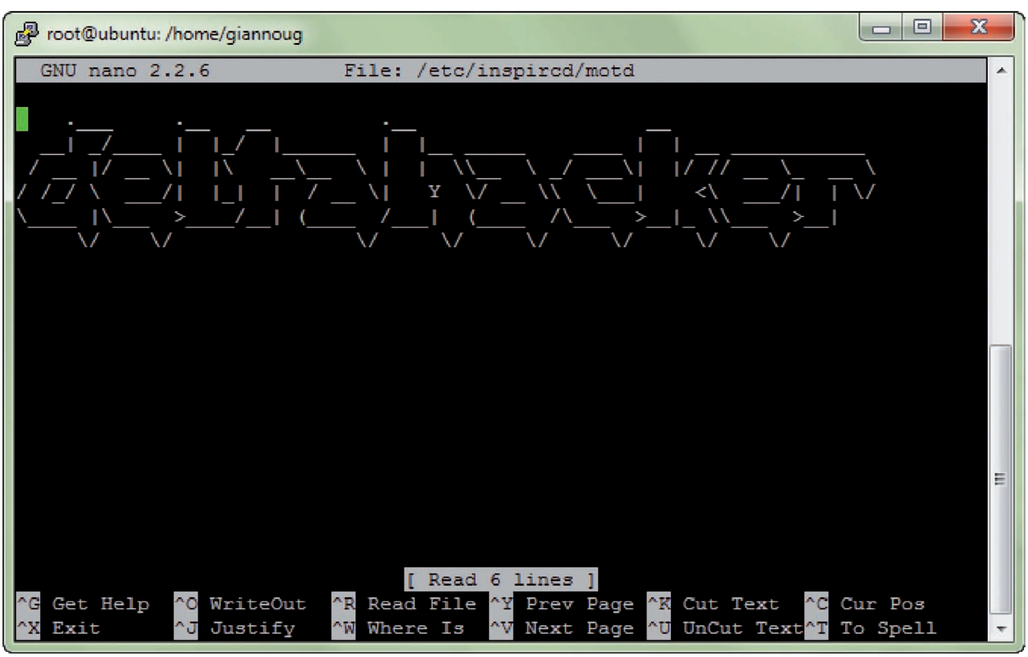
Με λίγη φαντασία και αρκετή υπομονή (όπως επίσης και με τη βοήθεια δικτυακών εργαλείων που παράγουν αυτόματα ascii art), φτιάξαμε ένα εντυπωσιακό MOTD. Ακριβώς αυτό θα βλέπουν όλοι όσοι συνδέονται στον IRC server μας!

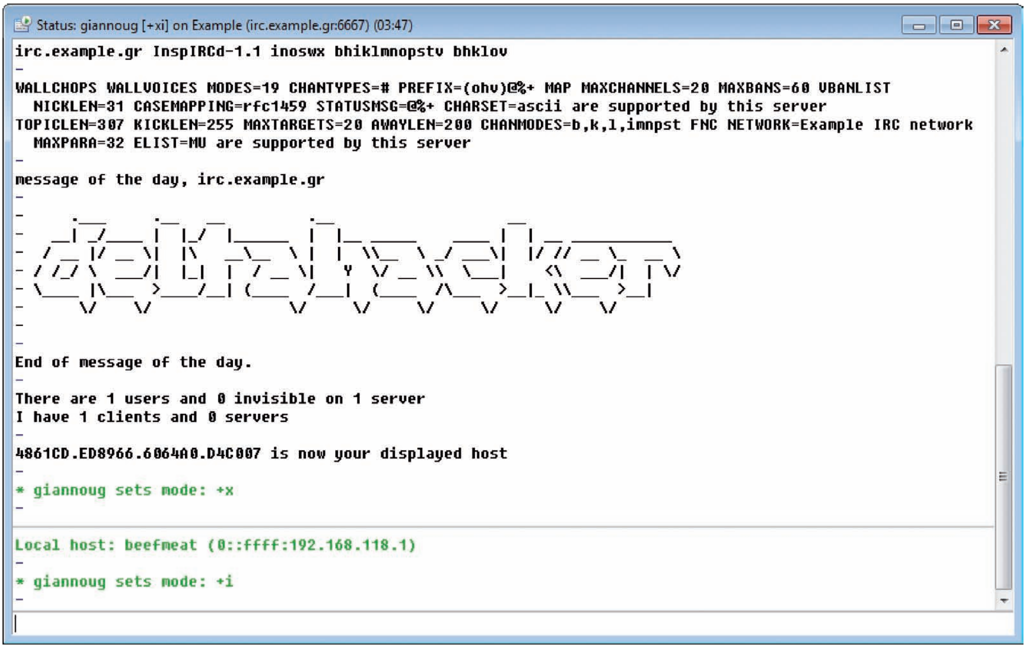
Ακολούθως, σε οποιαδήποτε επόμενη γραμμή, πρέπει να δώσουμε μερικές ακόμα πληροφορίες. Συγκεκριμένα, πρέπει να καθορίσουμε το κανάλι (ή τα κανάλια) στο οποίο θα συνδέονται αυτόματα οι χρήστες. Κάτι τέτοιο γίνεται με μια γραμμή σαν αυτή:

```
<autojoin channel="#deltahacker, #help">
```

Ο server μας έχει δύο κύρια κανάλια: το #deltahacker και το #help. Σε αυτά θα συνδέονται όλοι κι εκεί θα ξεσπάνε τα flame wars, το ένα μετά το άλλο ;)

Σημειώστε ότι για να ενεργοποιηθεί αυτή η ρύθμιση, όπως και όλες όσες





Το message of the day φαίνεται αρκετά όμορφο, δεν συμφωνείτε;

θα δούμε στη συνέχεια, πρέπει κάθε φορά να κάνουμε επανεκκίνηση του daemon. Όπως είδαμε νωρίτερα, αυτό γίνεται εκτελώντας το ακόλουθο:

```
sudo service inspircd restart
```

Προστασία των χρηστών

Ένα από τα μειονεκτήματα του IRC (τουλάχιστον για όσους φοβούνται υπερβολικά ή έχουν να κρύψουν πολλά) είναι το γεγονός ότι το hostname των χρηστών εμφανίζεται πάντα. Με απλά λόγια, οποιοσδήποτε χρήστης μπορεί να δει τη διεύθυνση IP οποιουδήποτε άλλου! Τι γίνεται σε περίπτωση που απλά δεν θέλουμε να εμφανίζεται κάτι τέτοιο για όλους τους χρήστες;

Η λύση δίνεται πάλι από ένα module, συγκεκριμένα το m_cloaking.so, το οποίο πετυχαίνει το λεγόμενο cloaking. Πρακτικά, αντικαθιστά το hostname του εκάστοτε χρήστη με την κρυπτογραφημένη εκδοχή του. Έτσι, αντί για τη διεύθυνση IP, εμφανίζεται ένα ακαταλαβίστικο string, το οποίο δεν προδίδει καθόλου την προέλευση του χρήστη. Όταν ενεργοποιούμε το συγκεκριμένο module, το inspIRCd υποστηρίζει αυτόματα ένα νέο mode για τους χρήστες. Τα user modes αποτελούν καταστάσεις στις οποίες μπορεί να βρίσκεται ένας χρήστης. Ουσιαστικά, αυτές οι καταστάσεις του προσδίδουν ή του αφαιρούν συγκεκριμένες δυνατότητες. Κάθε χρήστης μπορεί να αλλάξει το δικό του mode και μόνο. Βέβαια, κανένας απλός χρήστης δεν μπορεί να τροποποιήσει το δικό του mode κατάλληλα, ώστε να γίνει διαχειριστής. Αυτό μπορούν να το κάνουν μόνο οι διαχειριστές! Ας επιστρέψουμε στο θέμα μας, όμως. Όταν ενεργοποιούμε το cloaking module, το inspIRCd αποκτά ένα νέο mode (+x) για τους χρήστες. Για όποιον χρήστη ενεργοποιεί αυτό το mode, η διεύθυνσή του θα κρύβεται!

Κατά τα γνωστά, για την ενεργοποίηση και τη ρύθμιση του cloaking

module, πρέπει να επέμβουμε στο αρχείο ρυθμίσεων του inspIRCd. Αρχικά, πρέπει να δηλώσουμε τη χρήση του συγκεκριμένου module, καθώς και μερικών ακόμα που απαιτούνται για τη λειτουργία του. Έτσι, στην κορυφή του configuration file θα προσθέσουμε τις εξής γραμμές:

```
<module name="m_md5.so">
<module name="m_conn_umodes.so">
<module name="m_cloaking.so">
```

Μερικές γραμμές παρακάτω, θα προσθέσουμε και μερικές γραμμές σαν τις ακόλουθες:

```
<cloak mode="compat-host"
     key1="0x00123456"
     key2="0x01789abc"
     key3="0x02def012"
     key4="0x03345678"
     prefix="DH-">
```

Εδώ δηλώνουμε τα κλειδιά που θα χρησιμοποιούνται από το module, για την κρυπτογράφηση της εκάστοτε διεύθυνσης. Σαν κλειδιά μπορούμε να ορίσουμε οποιαδήποτε δεκαεξαδική τιμή, με μόνη προϋπόθεση να είναι μικρότερη από το 0x08000000. Αυτό ήταν! Τώρα, οι χρήστες του IRC server μας μπορούν να νιώθουν πιο ασφαλείς. Αφού αποθηκεύσουμε το αρχείο ρυθμίσεων κι επανεκκινήσουμε το inspIRCd, όποιος χρήστης θέλει να κρύψει την "καταγωγή" του αρκεί να εκτελέσει κάτι τέτοιο:

```
/mode my_nickname +x
```

Φυσικά, στη θέση του my_nickname, ο καθένας θα πρέπει να βάλει τα δικό του nickname.

Θωράκιση του server

Ο μικρός μας server είναι αξιοπρεπέστατος και μπορεί άνετα να παίξει το ρόλο του[3]. Υπάρχει ωστόσο μία ακόμα ρύθμιση, η οποία αν και δεν αλλάζει σε τίποτα όσα βλέπει ο τελικός χρήστης, προστατεύει τον server από διάφορες επιθέσεις. Αυτό που έχουμε κατά νου είναι η αποθήκευση των κωδικών σε κρυπτογραφημένη μορφή κι όχι σε μορφή απλού κειμένου. Έτσι, ακόμα κι αν καταφέρει κανείς να αποκτήσει shell access στον server μας, θα είναι πρακτικά αδύνατο να μάθει τους κωδικούς των χρηστών! Για το σκοπό αυτό θα αξιοποιήσουμε πάλι ένα από τα modules του inspIRCd. Αφού ανοίξουμε το αρχείο ρυθμίσεων, αρκεί να προσθέσουμε στην αρχή του την ακόλουθη γραμμή:

```
<module name="m_sha256.so">
```

Πλέον, για κάθε χρήστη που ορίζουμε εντός του αρχείου ρυθμίσεων, μπορούμε να χρησιμοποιούμε μια ελαφρώς διαφορετική σύνταξη. Για παράδειγμα, δείτε τη βελτιωμένη (και πολύ πιο ασφαλή) δήλωση του χρήστη giannoug:

```
<oper name="giannoug"
      hash="sha256"
      password="646801d8f659f8fc7f00696b05c281960fd49c2cf69592e3
29f65d96fda95536"
      host="*"
      type="NetAdmin">
```

3. Δηλαδή να φιλοξενήσει τις χαβαλεδιάρικες συζητήσεις μας ;)

Προσέξετε ότι μετά από το username ακολουθεί η δήλωση του μεγέθους hash. Εκεί προσδιορίζουμε τον αλγόριθμο κρυπτογράφησης, με τον οποίο δημιουργούμε το hash, για κάθε κωδικό. Σημειώστε ότι αντί για sha256, θα μπορούσαμε να είχαμε χρησιμοποιήσει το md5. Σε αυτή την περίπτωση, ωστόσο, θα έπρεπε να είχαμε ενεργοποιήσει και το αντίστοιχο module (m_md5.so). Ακριβώς από κάτω, έχουμε δώσει τον κωδικό μας (heavy_melon) σε κρυπτογραφημένη κατά sha256 μορφή. Όποιος και να δει το αρχείο ρυθμίσεων, θα του είναι αδύνατο να οδηγηθεί στον κωδικό μας!

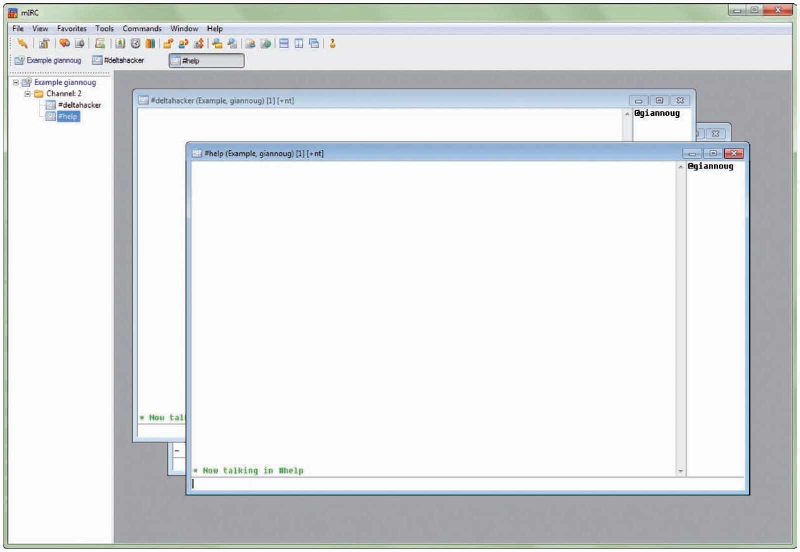
Σε αυτό το σημείο, όμως, πρέπει να αναφέρουμε κάτι ακόμα. Πώς μπορεί να κρυπτογραφήσει κανείς τον κωδικό του, κατά sha256; Ευτυχώς, η γραμμή εντολών του Linux διαθέτει εξ ορισμού ένα σχετικό εργαλείο. Για την κρυπτογράφηση οποιουδήποτε string κατά sha256, αρκεί να δώσουμε κάτι τέτοιο:

```
echo -n my_string_to_be_encrypted | sha256sum
```

Μόλις δώσουμε [Enter], στην επόμενη γραμμή θα εμφανιστεί η κρυπτογραφημένη εκδοχή του μηνύματός μας!

Καθολική αναβάθμιση

Όλα τα μεγάλα δίκτυα IRC παρέχουν αρκετές πρόσθετες υπηρεσίες (ChanServ, NickServ, OperServ, MemoServ κ.ά.) οι οποίες κάνουν τη ζωή χρηστών και διαχειριστών ευκολότερη. Ο δικός μας server δεν φιλοδοξεί να φιλοξενήσει χιλιάδες χρήστες και συζητήσεις, αλλά αυτό δεν σημαίνει ότι πρέπει να είναι φτωχότερος σε δυνατότητες. Για όσους θέλουν να εμπλουτίσουν τη λειτουργικότητα του server τους, μπορούν να στραφούν σε διάφορα σχετικά προγράμματα. Ένα από αυτά, το οποίο συνεργάζεται άψογα με τον InspIRCd, είναι το Atheme (www.atheme.net). Το Atheme προσφέρει μια πληθώρα δυνατοτήτων, όπως καλύτερη διαχείριση των καναλιών και των χρηστών, αποστολή μηνυμάτων offline και άλλα. Εναλλακτικά, μια εξίσου καλή επιλογή αποτελεί το Anope (www.anope.org).



Με το που συνδεόμαστε στον server, εισερχόμαστε αυτόματα και στα δύο προεπιλεγμένα κανάλια!



Skill: Intermediate

Tags: Python, Pygame, collision detection

Python Game Programming:

Η Εισβολή Αρχίζει!

by Sonic

Καθώς φαίνεται, τα ψέμματα τελείωσαν. Έχουμε το σκάφος μας, το κινούμενο φόντο και το laser, ενώ στο άρθρο 6.5 μας ήρθαν μια βόλτα και οι πρώτοι εξωγήινοι (http://deltahacker.gr/?p=6108). Λίγο διστακτικοί εμείς, τους κοιτάξαμε, τους πυροβολήσαμε και διαπιστώσαμε ότι δεν γίνεται τίποτα! Μα, τι λείπει επιτέλους από αυτό το παιχνίδι;

Αν είχατε ξεκινήσει το programming παιχνιδιών πριν μερικά χρόνια --τότε που η δημιουργία frame based shooters ήταν απλώς άπιαστο όνειρο-- και περιγράφατε το πρόβλημα στο φίλο σας που έγραφε εκείνη την εποχή το δικό του Invaders, θα σας έλεγε δύο λέξεις: collision detection. Η ανίχνευση συγκρούσεων είναι ένα βασικό θέμα (και πρόβλημα) σε όλα τα παιχνίδια δράσης. Είναι το τμήμα του προγράμματος που αναγνωρίζει πότε ο εξωγήινος δέχτηκε τη βολή, πότε το διαστημόπλοιο έσκασε πάνω στον κινούμενο μετεωρίτη, πότε ο Pacman έφαγε το φαντασματάκι.

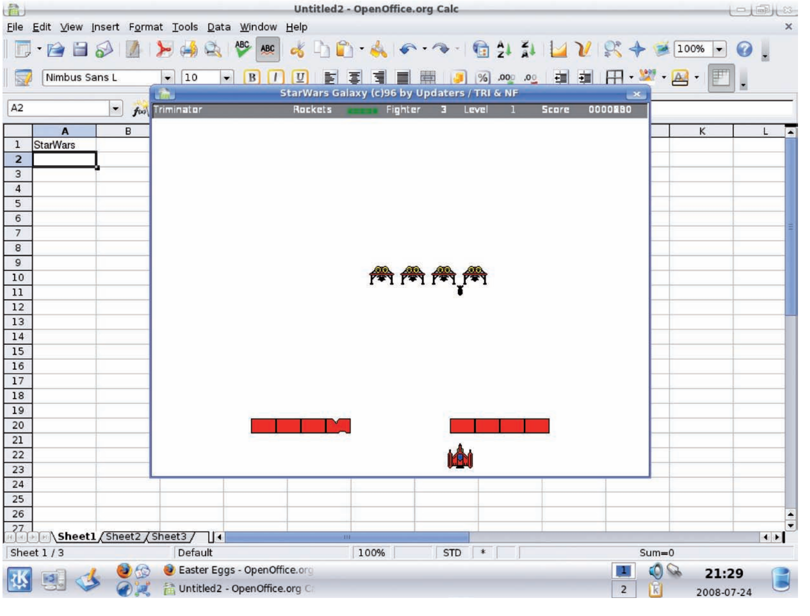
Μια κακοσχεδιασμένη ρουτίνα ανίχνευσης συγκρούσεων οδηγεί σε τραγελαφικά αποτελέσματα, όπως...

- Χαρακτήρες που περνάνε μέσα από τοίχους (μόνο ο κώδικας μας και τα φαντάσματα το καταφέρνουν αυτό)
- Βολές που ακουμπάνε --ή ακόμα χειρότερα περνάνε μέσα από-- χαρακτήρες, χωρίς να τους καταστρέφουν. Και καλά να είναι το δικό μας διαστημόπλοιο, τι γίνεται όμως με τυχόν απέθαντους εξωγήινους;
- Lasers που καταστρέφουν πράγματα τα οποία δεν φαίνονται να έχουν αγγίξει ή --ακόμα χειρότερα-- βρίσκονται στην άλλη άκρη της οθόνης.

Σε ένα παιχνίδι που *δεν* έχει γραφτεί με τη frame based λογική που χρησιμοποιούμε στη σειρά μας, η ανίχνευση συγκρούσεων δεν είναι και το πιο εύκολο πράγμα: Φανταστείτε ότι αν η γλώσσα προγραμματισμού παρέχει αυτόνομα κινούμενα sprites, το πρόγραμμά μας μπορεί να εκτελεί οποιαδήποτε τυχαία εντολή την ώρα που θα συγκρουστούν. Αν είμαστε τυχεροί, η εντολή που θα εκτελείται θα είναι αυτή που θα εντοπίζει τη σύγκρουση δύο sprites.

Για να αντιμετωπιστεί το πρόβλημα του collision detection, σε κάποιες διαλέκτους BASIC της Παλιάς Εποχής (TM) άρχισαν να εμφανίζονται

Τα easter eggs, αν δεν τα ξέρετε ήδη, είναι μικρά προγραμματάκια κρυμμένα μέσα σε μεγαλύτερα --και πολλές φορές «σοβαρά»-- προγράμματα, τα οποία ενεργοποιούνται με κάποιο περίεργο, κρυφό τρόπο. Στις παλιές εκδόσεις του OpenOffice υπήρχε κρυμμένο μέσα του ένα παιχνίδι με τίτλο StarWars, το οποίο στην πραγματικότητα αποτελούσε άλλη μία εκδοχή του Space Invaders. Για να ξεκινήσει έπρεπε κάποιος να γράψει =GAMES ("StarWars") στο πρώτο κελί ενός φύλλου του Calc. Δυστυχώς, στις νέες εκδόσεις των OpenOffice και LibreOffice τα easter eggs αφαιρέθηκαν...



εντολές με τη λογική των events (συμβάντων): Το πρόγραμμα εκτελούταν κανονικά, η γλώσσα έλεγχε αυτόματα κατά τακτά διαστήματα για συγκρούσεις κι έστελνε ένα event στο πρόγραμμα, το οποίο μετά επεξεργαζόμασταν για να διαπιστωθεί τι ...σκοτώθηκε. Όλα αυτά βέβαια για τους λίγους τυχερούς με τους αντίστοιχους υπολογιστές (εμείς ήμασταν καταδικασμένοι να πυροβολούμε δέκα φορές το κάθε καταραμένο UFO, μέχρι να εκτελεστεί η σωστή εντολή!)

Και τώρα που έχουμε frames, τι γίνεται; Μα, τα πράγματα έχουν απλοποιηθεί πολύ! Πάμε να δούμε τι γίνεται...

Collision Detection, Part I: Εξωγήινε σε έφαγα!

Ρίξτε λίγο μια ματιά στο super class ονόματι Craft και θα θυμηθείτε ότι έχουμε αποθηκεύσει τις συντεταγμένες του σκάφους (είτε του δικού μας είτε του εξωγήινου) σε ένα αντικείμενο τύπου Rect:

```
self.rect = pygame.Rect(coord,(self.ship_width, self.ship_
height))
```

Το παραπάνω μας έχει ήδη διευκολύνει στην κίνηση, καθώς μπορούμε με ένα αντικείμενο rect να αλλάξουμε τις συντεταγμένες επί τόπου, όπως φαίνεται στη ρουτίνα κίνησης:

```
distance_x = speed_x * time

distance_y = speed_y * time

self.rect.move_ip(distance_x,distance_y)
```

Δεν είναι όμως μόνο αυτό: Η κλάση Rect μας παρέχει τη συνάρτηση collidepoint, η οποία ελέγχει αν κάποιες συντεταγμένες που δίνουμε βρίσκονται μέσα στο παραλληλόγραμο (rectangle) που ελέγχουμε. Έτσι, αν AlienShip είναι ένα αντικείμενο τύπου «εξωγήινος» και μια βολή βρίσκεται στις συντεταγμένες (120,248), η εντολή

```
collision = AlienShip.rect.collidepoint((120,248))
```

θα επιστρέψει True αν υπάρχει «σύγκρουση», επιτρέποντάς μας να κάνουμε αυτό που πάντα θέλαμε: Να ξεπαστρέψουμε τους εισβολείς! (Δυστυχώς, με μια μικρή παραλλαγή θα επιτρέπεται και σε αυτούς να ξεπαστρεύουν εμάς.) Μια αρχική εκδοχή του κώδικα μπορεί να είναι η ακόλουθη:

```
for theshot in firelist:
    theshot.Move(time)
    theshot.Show(screen)
    if theshot.GoneAbove(0):
        firelist.remove(theshot)
    else:
        for AlienShip in AlienShips:
            if AlienShip.rect.collidepoint(theshot.GetXY()):
                explosion.play()
            if theshot in firelist:
                firelist.remove(theshot)
            AlienShips.remove(AlienShip)
```

Το πρώτο κομμάτι του κώδικα το ξέρετε ήδη: Είναι αυτό που κινεί τις βολές και ελέγχει αν κάποια έχει ξεφύγει από την οθόνη για να τη διαγράψει από τη λίστα. Αν δεν έχει ξεφύγει:

- Διατρέχουμε όλη τη λίστα των AlienShips
- Ελέγχουμε αν η τρέχουσα βολή (theshot) συγκρούεται με κάποιο AlienShip (εντολή collidepoint)
- Αν αυτό συμβαίνει, ακούγεται ο ήχος της έκρηξης (σε πείσμα της φυσικής και του κενού του διαστήματος), η βολή αφαιρείται από τη λίστα firelist και ο αντίστοιχος εξωγήινος από τη λίστα AlienShips. Η διαγραφή από τη λίστα σημαίνει και εξαφάνιση τους από την οθόνη στο αμέσως επόμενο καρέ!

Η συνάρτηση GetXY είναι πολύ απλή. Ανήκει στο Laser class και απλώς επιστρέφει τις συντεταγμένες της βολής μας:

```
def GetXY(self):
    return (self.x1,self.y1)
```

Δεν σχολιάζουμε καθόλου το explosion.play(), το οποίο είναι ένα ακόμα ηχητικό εφέ που προσθέσαμε.

Και ναι, αυτό είναι όλο! Αν τρέξετε το πρόγραμμα μπορείτε να πάρετε επιτέλους εκδίκηση, εξαφανίζοντας όσα κύματα εξωγήινων θέλετε. Όσο προλαβαίνετε, δηλαδή, μιας και δεν έχουν ακόμα αρχίσει να ρίχνουν...

Το ωραίο είναι ότι με τον τρόπο που φτιάξαμε το παιχνίδι, κάθε φορά που τελειώνουμε με ένα κύμα εξωγήινων, εμφανίζεται αμέσως το επόμενο. Θυμηθείτε το μαγικό κόλπο:

```
if not AlienShips:
    ...
```

Η αυτοκρατορία (των χαζών) αντεπιτίθεται!

Ένα παιχνίδι που θα πυροβολούμε μόνο εξωγήινους χωρίς να μας αντεπιτίθενται, θα γίνει βαρετό πολύ σύντομα: Κάτι σαν το Graphics Match του τεύχους 008, στο οποίο μόνο χάναμε :S Ήρθε λοιπόν η ώρα αυτά τα χαζά πλάσματα να αποκτήσουν λίγη αξιοπρέπεια, προβάλλοντας μια κάποια αντίσταση.

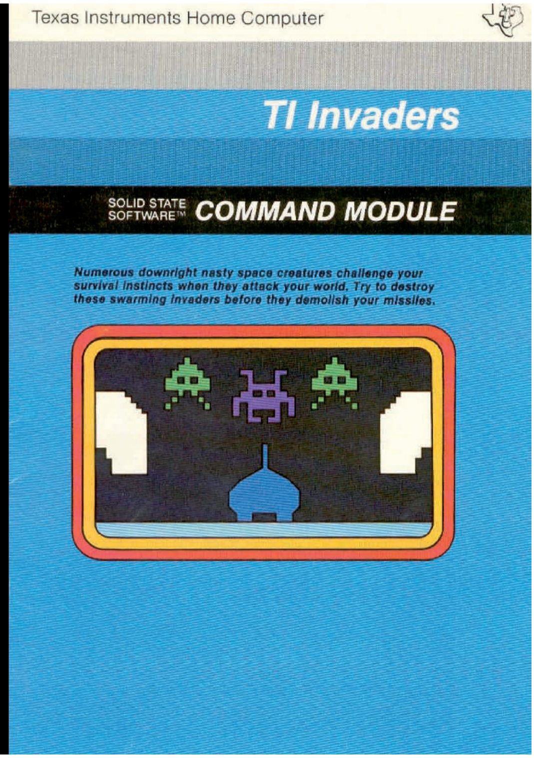
Πριν ξεκινήσουμε, ας θυμηθούμε λίγο τις παραμέτρους στον constructor του Laser class:

```
class Laser:
    def __init__(self, coord, color, size, speed, refline,
voffset):
```

όπου φυσικά coord είναι οι συντεταγμένες, color είναι το χρώμα της βολής (διαφορετικό για τον εξωγήινο), size είναι το μήκος της γραμμής Laser, refline είναι η γραμμή αναφοράς εκκίνησης της βολής (το κάτω μέρος για τον εξωγήινο) και voffset είναι μια μικρή διόρθωση στον κάθετο άξονα της βολής (που μάλλον δεν θα χρειαστούμε για τα AlienShip). Η συνάρτηση Fire στο Alien class, θα είναι κάπως έτσι:

```
def Fire(self):
    theshot = Laser((self.rect[0]+self.ship_midwidth, self.
rect[1]+self.firebaseline),self.firecolor, self.shot_height, self.
```






```
firespeed,self.rect[1]+self.firebaseline, 0)
                return theshot
```

Εννοείται ότι έχουμε ορίσει χρώμα, firebaseline και shot_height, στον constructor του Alien class.

Καλά όλα αυτά, αλλά δεν σημαίνει ότι οι εξωγήινοι ρίχνουν κιόλας. Θα πρέπει με κάποιο τρόπο να αποφασίζουν πότε (κι αν) θα ρίξει ο καθένας τους. Πάντως το σίγουρο είναι ότι δεν πρόκειται να βγάλουν χέρια για να χτυπάνε το [Space], στο πληκτρολόγιο. Για άλλη μια φορά, θα μας σώσει η random:

```
for AlienShip in AlienShips:
    AlienShip.Show(screen)
    AlienShip.Move(time)
    if randint(0,10)==9:
        if alienfirelist:
            if alienfirelist[-1].DistanceTravelled()>=100:
            alienfirelist.append(AlienShip.Fire())
            else:
            alienfirelist.append(AlienShip.Fire())
```

Ο εξωγήινος ρίχνει μόνον αν το randint(0,10) φέρει την τιμή 9. Ναι, ουσιαστικά ο εξωγήινος ρίχνει ζάρι, για να αποφασίσει αν θα ρίξει ή όχι. Και

Στο άρθρο του deltaHacker 010 ο γράφων έδειξε, μεταξύ άλλων, τη δική του εκδοχή του TI Invaders, γραμμένη σε BASIC, η οποία είχε γραφτεί πριν ακόμα μάθει ότι η TI είχε κυκλοφορήσει το Space Invaders σε cartridge. Οι δύο φωτογραφίες είναι από το παιχνίδι, ενώ βλέπετε και τις οδηγίες του. Είναι αρκετά δύσκολο και γίνεται ακόμα χειρότερο αν προσπαθείτε να παίξετε με τα original TI joysticks: Έχουν ψηφιστεί μέσα στα δέκα χειρότερα όλων των εποχών!

βέβαια μπορούμε να κάνουμε το παιχνίδι πιο δύσκολο γράφοντας κάτι σαν αυτό

```
if randint(0,10)>4:
        ...
```

οπότε και οι βολές θα είναι πιο συχνές. Το alienfirelist, φυσικά, είναι η λίστα που κρατάει τις εχθρικές βολές και αρχικοποιείται έξω από τον βρόχο με την εντολή:

```
alienfirelist = [ ]
```

Ό,τι ισχύει για τις δικές μας βολές ισχύει και για τις εχθρικές: Δεν ξεκινάει νέα βολή αν η προηγούμενη δεν έχει διανύσει μια ελάχιστη απόσταση που έχουμε ορίσει ως 100 pixels -- και ελέγχουμε με την DistanceTravelled(). Δυστυχώς, η DistanceTravelled() όπως την έχουμε φτιάξει μέχρι στιγμής δεν λειτουργεί σωστά για τις εχθρικές βολές. Μπορείτε να φανταστείτε γιατί;

Απλά, οι βολές αυτές πηγαίνουν προς τα κάτω, ενώ οι δικές μας προς τα πάνω! Έτσι η συνάρτηση παράγει αρνητικές τιμές για τις εχθρικές βολές, ενώ στην ουσία μας ενδιαφέρει η απόλυτη τιμή αυτής της απόστασης. Και μόλις είπαμε τη λέξη κλειδί για το πρόβλημά μας: η Python παρέχει τη συνάρτηση απόλυτης τιμής abs, όπως σχεδόν όλες οι γλώσσες προγραμματισμού! Άρα, η DistanceTravelled γίνεται:

```
def DistanceTravelled(self):
        return abs(self.refline – self.y1)
```

Η αλλαγή αυτή δεν επηρεάζει καθόλου τη λειτουργία της συνάρτησης,

Ο ZX-81, δημιούργημα της Sinclair Research, γνωστή κυρίως από τον ZX Spectrum, είναι ένα από τα πρώτα οικιακά micros με 1KB RAM (και εξωτερική επέκταση στα 16KB), με το καταπληκτικό χαρακτηριστικό ότι ήταν ...ασπρόμαυρο. Με ταχύτητα 1MHz ή 2MHz (στο λεγόμενο «fast mode») ήταν σε θέση να τρέχει το Space Invaders (προφανώς, η συγκεκριμένη εκδοχή είναι γραμμένη σε Assembly κι όχι σε BASIC).



στις δικές μας βολές. Το μόνο που μένει είναι να δείξουμε και να κινήσουμε τις βολές:

```
for theshot in alienfirelist:
    theshot.Move(time)
    theshot.Show(screen)
    if theshot.GoneBelow(laserdownlimit):
            alienfirelist.remove(theshot)
```

Αντι για GoneAbove έχουμε φυσικά GoneBelow, καθώς οι εχθρικές βολές κατευθύνονται προς τα κάτω (ή προς τα πάνω μας). Το laserdownlimit το έχουμε ορίσει έξω από το βασικό βρόχο, ως

```
laserdownlimit = screenheight - 40
```

Κανονικά θα έπρεπε να έχουμε και laseruplimit = 0, αντί να είναι hardcoded μέσα στον κώδικα (φτιάξτε το). Είμαστε τώρα ένα βήμα πριν οι εξωγήινοι γίνουν επικίνδυνοι: Ρίχνουν, αλλά είμαστε ακόμα άτρωτοι στις βολές τους. Αυτό θ' αλλάξει άμεσα!

Collision Detection, Part II: Οι βολές πέφτουν βροχή!

Εξασκηθείτε λίγο όσο οι εξωγήινοι είναι ακίνδυνοι: Σε πολύ λίγο θα γράψουμε το collision detection για τις βολές που πέφτουν στο *δικό* μας σκάφος! Ο σχετικός κώδικας είναι αρκετά όμοιος με το part I:

```
for theshot in alienfirelist:
    theshot.Move(time)
    theshot.Show(screen)
    if theshot.GoneBelow(laserdownlimit):
        alienfirelist.remove(theshot)
    else:
        if SpaceShip.rect.collidepoint(theshot.GetXY()):
            destroyed.play()
            if theshot in alienfirelist:
                alienfirelist.remove(theshot)
```

Για κάθε εχθρική βολή που ταξιδεύει στην οθόνη μας...

- Αν έχει βγει εκτός οθόνης εξαφανίζεται (αφαιρείται από τη λίστα alienfirelist)
- Διαφορετικά, αν έχει χτυπήσει το διαστημόπλοιο μας ακούγεται ο ήχος της έκρηξης (destroyed) και η βολή πάλι αφαιρείται από τη λίστα
- Αν δεν συμβαίνει τίποτα από τα δύο, η βολή απλώς συνεχίζει να κινείται

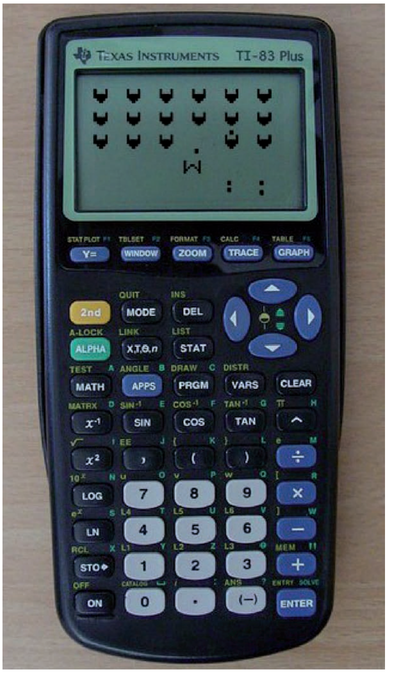
Παρατηρήστε βέβαια ότι το διαστημόπλοιο μας δεν διαλύεται με τη βολή. Για την ακρίβεια σε λίγο θα έχουμε ασπίδα και θα χάνουμε όταν η ισχύς της πέφτει στο μηδέν. Όσο πάντως πετυχαίνουμε εξωγήινους, η ασπίδα μας σιγά σιγά επανέρχεται. Τελικά, είναι μεγάλο πλεονέκτημα να είσαι προγραμματιστής: Ορίζεις εσύ τους κανόνες του παιχνιδιού -- προς το συμφέρον σου, βέβαια!

Για την ώρα πάντως δεν έχουμε ούτε ασπίδα, ούτε καν score. Α, έχουμε όμως κι ένα ακόμα προβληματάκι: Δεν υπάρχει καμιά ένδειξη στην οθόνη για το ότι έχουμε χτυπηθεί, μόνο ένας ήχος ακούγεται. Όλα αυτά πρέπει, φυσικά, να τα φτιάξουμε.

Τήρηση score

Αυτό είναι εύκολο. Ορίστε μια πολύ απλή κλάση για το ScoreBoard:

```
class ScoreBoard:
     def __init__(self,x,y,font, fontsize):
                     self.x = x
                     self.y = y
                     self.font = pygame.font.
SysFont(font,fontsize)
                     self.score = 0
def Change(self, amount):
             self.score += amount
def Show(self,surface):
             scoretext = self.font.render("Score: "+str(self.
score), True, (0,0,255))
             surface.blit(scoretext,(self.x,self.y))
def GetValue(self):
             return self.score
def SetValue(self, score):
             self.score = score
```



Τα παιχνίδια τύπου Invaders είναι τόσο δημοφιλή που έχουν γραφτεί για τα πιο απίθανα μηχανήματα. Στη φωτογραφία βλέπετε το Space Invaders για την προγραμματιζόμενη επιστημονική αριθμομηχανή (sic) TI-83, της Texas Instruments. Και τώρα ξέρετε τι δώρο να ζητήσετε από τους γονείς σας, ως πρωτοετής φοιτητής. Για καθαρά εκπαιδευτικούς σκοπούς, βέβαια!

Το τι γίνεται είναι πιστεύουμε εντελώς προφανές. Το Score θα εμφανίζεται στη θέση (x,y) της οθόνης, με τη γνωστή μας μέθοδο blit. Έχουμε τις συναρτήσεις Change, για να προσθέσουμε στο score την τιμή amount, και GetValue / SetValue, για να διαβάσουμε την τιμή ή να θέσουμε το score απευθείας σε μια τιμή της αρεσκείας μας (πάλι στο cheat πάει το μυαλό σας ε;). Στο κύριο πρόγραμμά μας θα έχουμε:

- Αρχικοποίηση Score...

```
score =
ScoreBoard(0,0,"impact",32)
```

- Σε κάθε επιτυχία μας...

```
if AlienShip.rect.
collidepoint(theshot.GetXY()):
       score.Change(10)
```

Για να δούμε όμως και αυτή την ασπίδα που λέγαμε... Είναι νομι-

ζουμε ώρα να σταματήσουμε να παριστάνουμε τους Highlanders!

ShieldMeter class

Το ShieldMeter class είναι αρκετά απλό. Σχεδιάζει μια μπάρα στην οθόνη που μειώνεται καθώς πέφτει η ενέργεια της ασπίδας μας (όσο δηλαδή καθόμαστε και τις τρώμε άγρια από τους εξωγήινους) ή αυξάνεται μετά από κάποιες δικές μας πετυχημένες βολές. Αν η ασπίδα μας πέσει κάτω από κάποια τιμή (warnvalue) το χρώμα της μπάρας αλλάζει. Προφανώς η ασπίδα έχει μια μέγιστη τιμή (maxvalue), πέρα από την οποία δεν αυξάνεται και εννοείται δεν μπορεί να πάει κάτω από το μηδέν (στο μηδέν βλέπουμε ήδη τα ψηφιακά ραπανάκια να φυτρώνουν από κάτω) (Σ.τ.Ε. Προσωπικά θα περίμενα ψηφιακά ραδίκια, αλλά τέλος πάντων.)

Την Παλιά Εποχή (TM), όταν βαριόμασταν το Space Invaders σειρά είχε το Pacman. Άπειρα joysticks πρέπει να έχουν διαλυθεί, εξαιτίας αυτών των δύο παιχνιδιών (Σ.τ.Ε. Απειραριθμήσιμα, για την ακρίβεια). Στη φωτογραφία βλέπετε ένα από τα περιοδικά που ασχολείται με retro gaming -- και πρέπει να σας πληροφορήσουμε ότι η retro σκηνή καλά κρατεί. Οι περισσότεροι βέβαια χρησιμοποιούν πλέον εξομοιωτές, αλλά όσοι έχουν τα original μηχανήματα δεν τα αλλάζουν με τίποτα!

```
class ShieldMeter:
    def __init__(self, x, y, maxvalue, warnvalue):
```

Το μόνο ενδιαφέρον κομμάτι στον κώδικα είναι η σχεδίαση με την draw.rect (οι άλλες συναρτήσεις είναι τόσο απλές όσο και αυτές του ScoreBoard):

```
def Show(self, surface):
    if self.currentvalue < self.warnvalue:
                self.shieldcolor = (255,0,0)
    else:
                self.shieldcolor = (0,255,0)
    pygame.draw.rect(surface,self.shieldcolor,(self.x, self.y,
self.currentvalue,25))
```

Αυτό το μαγικό 25, είναι το ύψος της μπάρας στην οθόνη. Μάλλον θα θέλετε να κάνετε κάτι γι' αυτό στον constructor, έτσι δεν είναι;



Στο κύριο πρόγραμμα, το shield αρχικοποιείται ως εξής:

```
shield =
ShieldMeter(200,10,250,75)
```

όπου τα 200, 10 είναι οι συντεταγμένες εμφάνισης, το 250 είναι η μέγιστη τιμή και το 75 η τιμή προειδοποίησης. Σε κάθε (τυχερή) βολή των χαζοεξωγήινων χάνουμε 25 μονάδες ασπίδας:

```
if SpaceShip.rect.
collidepoint(theshot.
GetXY()):
    destroyed.play()
    shield.Decrease(25)
```

Κάθε φορά που το score μας διαιρείται ακριβώς με το 100, παίρνουμε 25 μονάδες ασπίδας:

```
if AlienShip.rect.collidepoint(theshot.GetXY()):
     score.Change(10)
     explosion.play()
     if score.GetValue() % 100 == 0:
                    shield.Increase(25)
```

Φυσικά, μέσα στον κύριο βρόχο υπάρχει η εντολή shield.Show(screen), προκειμένου να εμφανίζεται η ασπίδα μας στην οθόνη!

Ανίχνευση τέλους παιχνιδιού

Όλα τα πράγματα έχουν ένα τέλος και μοιραία κάποια στιγμή οι εξωγήινοι θα μας φάνε λάχανο! Το παιχνίδι προφανώς πρέπει να ανιχνεύει το τέλος της ασπίδας, να δείχνει μήνυμα Game Over και να μην επιτρέπει τη συνέχεια. Για το μήνυμα έχουμε μια πολύ απλή συνάρτηση GameOverShow, που κάνει blit τα αντίστοιχα μηνύματα:

```
def GameOverShow(screen):
     font = pygame.font.SysFont("impact", 32)
     gameovertext = font.render("Game Over!",True,(255,255,255))
     text_x = CenterMessage(screen, gameovertext)
     screen.blit (gameovertext,(text_x,280))
     gameovertext = font.render("Press R to Restart", True,
(255,255,255))
     text_x = CenterMessage(screen, gameovertext)
     screen.blit(gameovertext,(text_x,320))
     return
```

Όπως βλέπετε, έχουμε προσθέσει και την Pygame εκδοχή της CenterMessage, για το κεντράρισμα του κειμένου. Στον κύριο βρόχο έχουμε:

```
if shield.GetValue() == 0:
     GameOverShow(screen)
     GameOver = True
```

Για να εξασφαλίσουμε ότι το παιχνίδι δεν παίζει όσο το GameOver = True, αλλά και να επιτρέψουμε την επανεκκίνηση του με το πλήκτρο [R], έχουμε αλλάξει κατάλληλα τις εντολές στην ανίχνευση του πληκτρολογίου:

```
if key[K_SPACE] and not GameOver:
```

Το διαστημόπλοιό μας ρίχνει μόνον αν δεν έχει επέλθει το GameOver. Κινείται όμως αριστερά – δεξιά (demo mode). Οι εξωγήινοι μάς ρίχνουν, αλλά οι βολές τους περνάνε από μέσα μας χωρίς να μας πειράξουν, επειδή έχουμε αλλάξει τη γραμμή...

```
if SpaceShip.rect.collidepoint(theshot.GetXY()) and not
GameOver:
```

Τέλος, πιέζοντας το πλήκτρο [R] ο χρήστης έχει δυνατότητα να ξαναπαίξει. Το παιχνίδι είναι τόσο εθιστικό άλλωστε, που θα ήταν αδύνατον να παραλείψουμε αυτό το feature :)

```
if key[K_r] and GameOver:
      GameOver = False
      shield.SetValue(250)
      score.SetValue(0)
```

Φυσικά, η μεταβλητή GameOver αρχικοποιείται έξω από το βρόχο, ως GameOver = False.

Κατεβάστε τον πλήρη κώδικα του παιχνιδιού από το

http://bit.ly/dh011pygameinv

Επίλογος (;)

Επιτέλους, μετά από επτά τεύχη κι άλλα τόσα online (συμπληρωματικά) άρθρα, έχουμε ένα πλήρως λειτουργικό παιχνίδι το οποίο θα μας διασκεδάσει με τις ώρες τόσο παίζοντας μαζί του, όσο κι επεκτείνοντάς το! Γιατί, φυσικά, ο σωστός προγραμματιστής διασκεδάζει περισσότερο γράφοντας το παιχνίδι, παρά παίζοντάς το. Εξάλλου, όσο παίζουμε ανακαλύπτουμε δυνατότητες και λειτουργίες που θέλουμε να προσθέσουμε: Ο κώδικας μάς έρχεται σε κύματα (σαν τους εξωγήινους, ένα πράγμα) και δεν τελειώνει ποτέ...



*Ανίχνευση συγκρούσεων με τον παλιό, καλό (;) τρόπο. Τα sprites κινούνται αυτόματα και η σύγκρουση δεν γίνεται αντιληπτή αν *δεν* συμπέσει με την εκτέλεση της εντολής ανίχνευσης. Στην TI Extended BASIC, η εντολή αυτή είναι η CALL COINC (από το coincidence, σύμπτωση) και φαίνεται στο listing. Δείτε και το ακόλουθο, διασκεδαστικό video, που δείχνει το πρόγραμμα και συγκρίνει το τότε με το τώρα: http://bit.ly/colldetect*

Με αυτό το άρθρο --για την ακρίβεια με το άρθρο 7.5 που θα βρείτε στο http://deltahacker.gr/?p=6141-- ολοκληρώνεται η σειρά προγραμματισμού παιχνιδιών σε Python και Pygame. Ελπίζουμε να διασκεδάσατε διαβάζοντας την. Ο γράφων πάντως είχε να διασκεδάσει τόσο γράφοντας κώδικα από την εποχή του TI-99!

Στο άρθρο 7.5 θα δούμε πώς θα κάνετε το διαστημοπλοιάκι ν' αλλάζει χρώμα όταν δέχεται βολές. Ακούγεται απλό, αλλά θα διαπιστώσετε ότι χρειάζεται να μάθετε και νέα κόλπα για να γίνει σωστά. Θα συζητήσουμε ακόμα άλλες βελτιώσεις που μπορείτε να κάνετε στον κώδικα αλλά και ιδέες για να επεκτείνετε το παιχνίδι, μια πρόκληση που αφήνουμε αποκλειστικά στα χέρια σας και στην ...καφεΐνη που θα καταναλώσετε.

Όπως θέλουμε να πιστεύουμε ότι αποδείξαμε, ο προγραμματισμός όχι μόνο δεν είναι βαρετός, αλλά αντίθετα αποτελεί την πιο ενδιαφέρουσα και δημιουργική ασχολία γύρω από τους υπολογιστές. Ελπίζουμε τα επιχειρήματα μας να ήταν πειστικά. Μην πείτε όχι, γιατί θα σας κυνηγήσουμε με το laser!



Πανίσχυρος και ικανότατος router,

για το οικιακό μας δίκτυο!

by giannoug

Skill: Beginner

Tags:
dd-wrt,
OpenWRT,
router, custom
firmware

Πρακτικά, όλοι έχουμε από ένα router στο σπίτι. Η δικτυακή αυτή συσκευή είναι σαν μια γέφυρα μεταξύ των υπολογιστών/δικτυακών συσκευών του σπιτιού και του Internet. Στη συντριπτική πλειονότητά τους, τα routers αυτά συνδέονται σε μια τηλεφωνική πρίζα κι αναλαμβάνουν να μεταφέρουν δεδομένα TCP/IP μέσα από τον τηλεφωνικό χαλκό, από και προς την εταιρεία που μας προσφέρει το Internet. Το πρωτόκολλο που χρησιμοποιούν γι' αυτό το κομμάτι της διαδρομής είναι συνήθως το *DSL.



Παρόλο που έχουμε συνηθίσει να τα λέμε "routers», στην πραγματικότητα πρόκειται για modem/routers. Πράγματι, πέρα από το να αποτελούν γέφυρες ή πύλες μεταξύ των υπολογιστών του τοπικού δικτύου (συνήθως του οικιακού) και κάποιου εξωτερικού δικτύου (συνήθως του Internet), στην πραγματικότητα έχουν και την «εξυπνάδα» ενός modem, π.χ., για να διαμορφώνουν το σήμα ώστε να ταξιδεύει μέσα από το τηλεφωνικό καλώδιο με βάση το πρωτόκολλο *DSL.

Μια αρκετά μεγάλη κοινότητα έχει αναπτυχθεί γύρω από τους routers, με σκοπό να τους κάνει αρκετά καλύτερους. Η κοινότητα αυτή έχει δημιουργήσει δύο διαφορετικές διανομές Linux, ικανές να μετατρέψουν έναν απλό, οικιακό router, σε έναν πλήρη, ικανότατο server. Όπως και με κάθε διανομή (και κοινότητα) Linux, έτσι κι εδώ η καθεμία έχει την δική της φιλοσοφία κι ακολουθεί διαφορετικό δρόμο από τις άλλες. Ονομαστικά, οι δύο αυτές διανομές είναι το dd-wrt (www.dd-wrt.com) και το OpenWRT (https://openwrt.org). Θα μιλήσουμε και για τα δύο, αλλά θα ασχοληθούμε εκτενέστερα με το dd-wrt.

Η αρχή έγινε με το routerάκι της φωτογραφίας. Το Linksys WRT54G ήταν ένα από τα πρώτα Linux routers με ελεύθερα διαθέσιμο τον κώδικα του firmware!

Σύντομη ιστορική αναδρομή

Όλα ξεκίνησαν από το Linksys WRT54G, ίσως το πρώτο Linux router με ελεύθερα διαθέσιμο τον κώδικα του firmware του. Εξαιτίας αυτού του γεγονότος, ανήσυχοι προγραμματιστές σήκωσαν τα μανίκια και στρώθηκαν στη δουλειά, αν μη τι άλλο για να δουν τι μπορούν να κάνουν με τον κώδικα. Έτσι, δεν άργησαν να εμφανιστούν τα πρώτα custom firmware για το συγκεκριμένο routerάκι, τα οποία φυσικά διατίθεντο δωρεάν στο Internet.



Το dd-wrt έχει δωρεάν αλλά και εμπορικό χαρακτήρα. Η επί πληρωμή εκδοχή του firmware προσφέρει επιπλέον δυνατότητες σε επαγγελματίες χρήστες και περιβάλλοντα (www.dd-wrt.com/site/professional/customization). Η δωρεάν εκδοχή, ωστόσο, υπερκαλύπτει όλους τους οικιακούς χρήστες!

Η αρχή είχε γίνει. Πλέον, κάθε router με ίδιο SoC (System on a Chip) ήταν ικανό να τρέξει τα custom firmware. Όλο και περισσότεροι κατασκευαστές αποφάσιζαν να βασιστούν στο Linux, νέες συσκευές έβγαιναν στην αγορά και, χάρη στην άδεια χρήσης GNU GPL, ο κώδικας του πυρήνα δημοσιευόταν επίσης. Το ένα custom firmware έφερε το άλλο, δημιουργήθηκαν οι πρώτες ομάδες και, κάπως έτσι, καταλήξαμε σήμερα με δύο μεγάλες διανομές Linux αποκλειστικά για routers.

Πλέον, η υποστήριξη hardware είναι αρκετά μεγάλη. Πάρα πολλά routers χρησιμοποιούν ακριβώς το ίδιο SoC, μοντέλα της ίδιας εταιρείας έχουν παρόμοιο hardware, routers διαφορετικών εταιρειών επίσης, οπότε η μεταφορά κάποιου custom firmware σε νέο router αποτελεί δουλειά ενός σαββατοκύριακου!

Δυστυχώς, τόσο το dd-wrt όσο και το OpenWRT δεν υποστηρίζουν modem/router αλλά σκέτα routers. Έτσι, αν θέλετε να βασίσετε το οικιακό σας δίκτυο στο dd-wrt ή στο OpenWRT, το modem/router που έχετε

από τον ISP σας θα συνεχίσει να σας συντροφεύει :) Απλά, απευθείας "πίσω" από το modem/router θα συνδέσετε τον router με το dd-wrt/OpenWRT, θα το βάλετε στο DMZ του modem/router κι από εκεί και πέρα θα σας απασχολεί μόνο το "σκέτο" αλλά παντοδύναμο router.

Το OpenWRT έχει άλλη λογική. Ναι μεν στοχεύει στην προσθήκη επιπλέον δυνατοτήτων, αλλά αυτό το καταφέρνει όχι μέσα από κάποιο εμπορικό μοντέλο αλλά προβάλλοντας τον ελεύθερο χαρακτήρα του Linux: Κάθε ενδιαφερόμενος αναλαμβάνει δράση και προσθέτει όποιο χαρακτηριστικό ή δυνατότητα θέλει, ενώ τα firmware που προκύπτουν είναι πάντα δωρεάν.

Η κύρια διαφορά του dd-wrt με το OpenWRT είναι ότι το πρώτο έχει χαρακτήρα εμπορικό. Συγκεκριμένες λειτουργίες του ξεκλειδώνουν με πληρωμή και περιέχει λειτουργίες από τρίτους κατασκευαστές, που συμβάλλουν στον εμπλουτισμό των δυνατοτήτων του.

Αγορά εξοπλισμού

Δυστυχώς, δεν υποστηρίζονται όλα τα routers Εκεί Έξω (TM). Δεν μπορούμε να πάμε σε ένα μαγαζί, να επιλέξουμε οποιοδήποτε και με το "έτσι θέλω" να του εγκαταστήσουμε κάποιο custom firmware. Στην πραγματικότητα υποστηρίζονται μονάχα μερικά routers -- αν και οι επιλογές είναι πάρα πολλές. Θα χρειαστεί να συμβουλευτούμε τον πίνακα με τις υποστηριζόμενες συσκευές του dd-wrt στο

www.dd-wrt.com/wiki/index.php/Supported_Devices

ή/και εκείνον με τις συσκευές στις οποίες παίζει το OpenWRT, στο

http://wiki.openwrt.org/toh/start

Στις αναζητήσεις σας θα βρείτε routers τα οποία δέχονται τόσο το dd-wrt όσο και το OpenWRT, αυτό όμως δεν είναι ο κανόνας.

Η καλύτερη επιλογή --τουλάχιστον για τον κύριο που πατάει τα πλήκτρα τώρα-- είναι τα router της TP-Link και πιο συγκεκριμένα τα μοντέλα TL-WR741, TL-WR841, TL-WR941 και TL-WR1043. Αυτά αποτελούν τα βασικά μοντέλα μιας αρκετά μεγάλης σειράς. Όσο μεγαλύτερο το μοντέλο, τόσα περισσότερα χαρακτηριστικά έχει. Μια εκτενέστερη λίστα μπορούμε να δούμε στο επίσημο wiki:

http://wiki.openwrt.org/toh/tp-link/start



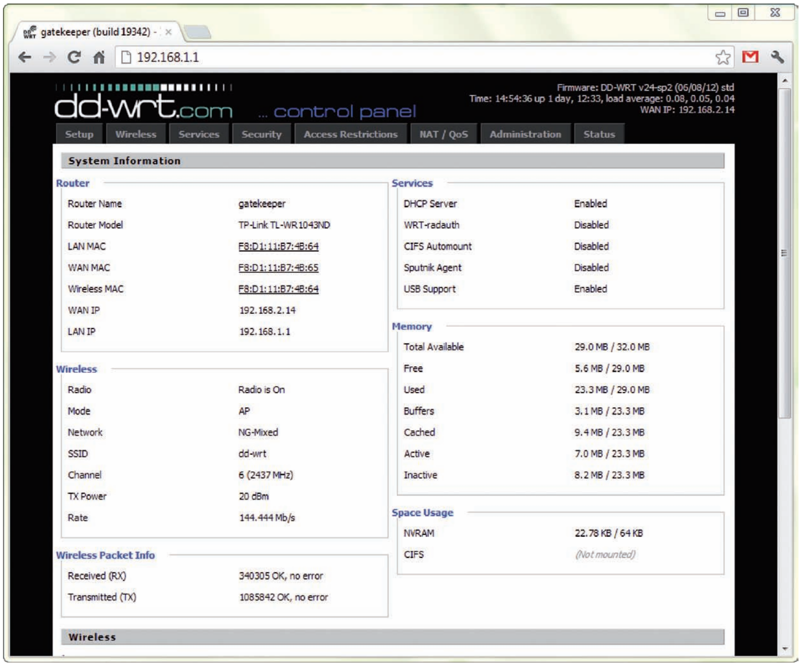
Για το άρθρο βασιστήκαμε στο εικονιζόμενο routerάκι. Πρόκειται για ένα TP-Link WR1043ND, με τίμη γύρω στα 55€. Περιλαμβάνει USB θύρα και Gigabit Ethernet, μαζί με WiFi 802.11n.

Οι τιμές ξεκινούν από τα 15€ για το μικρό TL-WR741 και φθάνουν έως τα 60€, για το TL-WR1043. Σε κάθε περίπτωση, είναι πολύ εύκολο να τα πειράξει κανείς. Στα πλαίσια της προετοιμασίας του παρόντος άρθρου παίζαμε με ένα TP-Link TL-WR1043ND, συνδεδεμένο σε ένα modem/router που είχε αγοραστεί πριν μερικούς μήνες. Έως τη στιγμή που γράφονται οι γραμμές αυτές, το TP-Link λειτουργεί με dd-wrt για αρκετές εβδομάδες και χωρίς προβλήματα. Ο λόγος που επιλέξαμε το "μεγάλο" μοντέλο είναι οι πρόσθετες δυνατότητές του, όπως η θύρα USB και το Gigabit Ethernet. Όσα συζητήσουμε στη συνέχεια πιθανώς να ισχύουν για όλες τις παρόμοιες συσκευές, δεδομένου ότι το dd-wrt τις αξιοποιεί με τον ίδιο τρόπο.

Εγκατάσταση

Για το παρόν κείμενο, εστιάζουμε στη συνέχεια την προσοχή μας στο dd-wrt. Κάθε συσκευή έχει τον δικό της, μερικές φορές ιδιότροπο, τρόπο

Η αρχική οθόνη του dd-wrt, πλήρης από χρήσιμες πληροφορίες όπως διαθέσιμη RAM/NVRAM, κατάσταση των services κ.λπ.



εγκατάστασης του dd-wrt. Στις περισσότερες περιπτώσεις η εγκατάσταση γίνεται περνώντας στη συσκευή δύο αρχεία, μέσα από το μενού αναβάθμισης του router. Έτσι ακριβώς γίνεται και στα routers της TP-Link, που αναφέραμε προηγουμένως. Καλό θα ήταν για αρχή να μην σκεφτούμε καν ν' ασχοληθούμε με άλλο τρόπο εγκατάστασης, αφού είναι πιθανό να τα βρούμε σκούρα. Κακά τα ψέμματα: Η εγκατάσταση μέσω του μενού αναβάθμισης είναι ο μόνος τρόπος που δεν απαιτεί κολλητήρι και ...εξωτερικό hardware! (Σας πείσαμε ή ακόμα; :D)

Πού θα βρούμε αυτά τα δύο αρχεία που αναφέραμε πριν; Χμ, εξαρτάται. Το TP-Link TL-WR1043ND, π.χ., υποστηρίζεται επίσημα από το dd-wrt. Και πάλι, όμως, χρειάζεται προσοχή, αφού υποστηρίζονται μόνο τα hardware revisions 1.1 και 1.4. Το μοντέλο πάλι που είχαμε εμείς, ήταν hardware revision 1.8[1]. Τι κάνουμε σε μια τέτοια περίπτωση; Σίγουρα δεν πανικοβαλλόμαστε, αφού με λίγο ψάξιμο πιθανώς θα βρούμε έκδοση συμβατή με το routerάκι μας. Απλά, αυτή δεν θα είναι από την επίσημη ομάδα του dd-wrt αλλά από κάποιον της κοινότητας, ο οποίος έχει ήδη μεταγλωττίσει τον κώδικα για το συγκεκριμένο board revision. Προσέξτε: Αυτό δεν σημαίνει ότι το dd-wrt δεν υποστηρίζει τον router μας, απλά ότι δεν μπορούν να εγγυηθούν πως θα δουλεύει 100% σωστά.

Πάντως τις περισσότερες φορές τα board revisions δεν διαφέρουν καθόλου μεταξύ τους. Ανά διαστήματα, οι εταιρείες κάνουν μικρές αλλαγές στην πλακέτα για να διορθωθούν προβλήματα ή να κάνουν μικροβελτιώσεις. Έτσι συνέβη και στην περίπτωση μας, όπου σύμφωνα με το wiki του OpenWRT[2] η πλακέτα με έκδοση 1.8 του TP-Link TL-WR1043ND είναι ίδια ακριβώς με την πλακέτα της έκδοσης 1.1[3].

1. Μπορούμε να το δούμε στο αυτοκόλλητο, κάτω από τον ίδιο τον router.
2. Μεταξύ των project dd-wrt και OpenWRT συχνά υπάρχει στενή συνεργασία.
3. Τότε γιατί χρειαζόμαστε διαφορετικό firmware και δε βάζουμε απλά το 1.1 ή το 1.4; Γιατί πιθανώς γίνεται

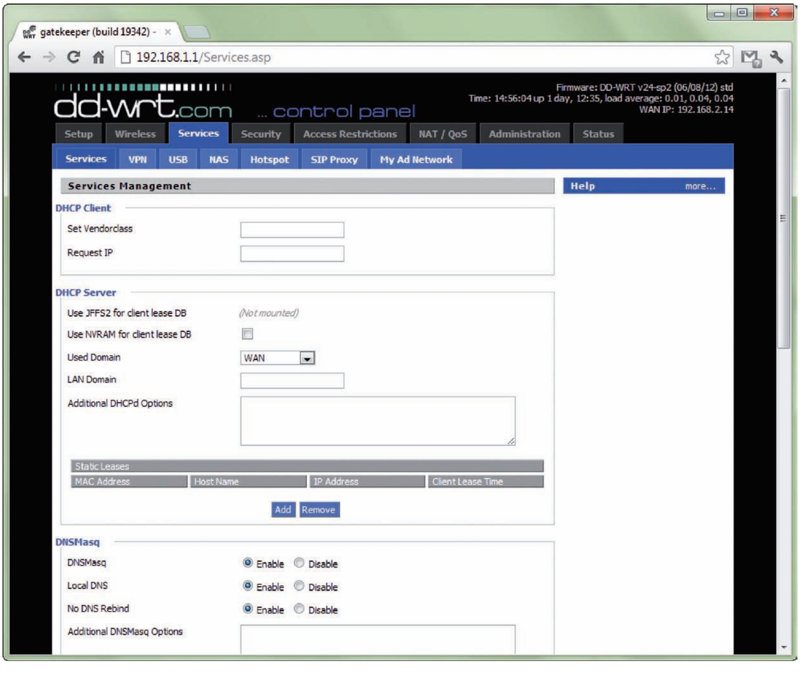
Έχουμε λοιπόν τα δύο αρχεία --στην περίπτωσή μας τα factory-to-ddwrt.bin και tl-wr1043nd-webflash.bin--, τα οποία και πήραμε από την router database του dd-wrt, από το επίσημο forum ή ακόμη και χρησιμοποιώντας το Google.

Το πρώτο αρχείο είναι αυτό που κυρίως μας ενδιαφέρει. Κάνει όλες εκείνες τις απαραίτητες προετοιμασίες για την εγκατάσταση του dd-wrt. Το δεύτερο αρχείο δεν μας χρειάζεται και χρησιμοποιείται σε περιπτώσεις που θέλουμε να αναβαθμίσουμε ένα ήδη πειραγμένο dd-wrt router.

Ανοίγουμε τη σελίδα διαχείρισης του router μας και πηγαίνουμε στο μενού αναβάθμισης. Για το TP-Link TL-WR1043ND --και γενικά για όλα τα συγγενικά router που αναφέραμε πριν-- η σελίδα βρίσκεται στο `http://192.168.1.1`. Στη συνέχεια επιλέγουμε το αρχείο factory-to-ddwrt.bin και πατάμε στο Upgrade. Το αρχείο θα μεταφερθεί στον router και θα ξεκινήσει η διαδικασία της αναβάθμισης. Σε καμία, μα καμία, περίπτωση δεν πειράζουμε οποιοδήποτε καλώδιο της συσκευής. Αν διακόψουμε τη διαδικασία υπάρχει μεγάλη περίπτωση να καταλήξουμε με ένα ακριβό door stopper! Η διαδικασία ανάκτησης από λανθασμένη αναβάθμιση, εκτός του ότι είναι χρονοβόρα απαιτεί την αγορά επιπλέον hardware. Προσοχή, λοιπόν!

Ένας νέος κόσμος

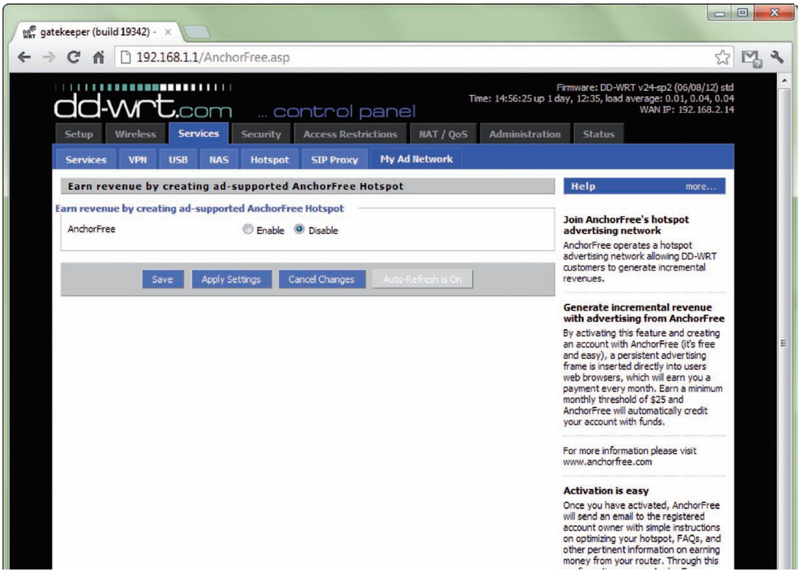
Μπορούμε να σας υποσχεθούμε πως το dd-wrt θα σας αφήσει με το στόμα ανοιχτό, ήδη στο πρώτο λεπτό της γνωριμίας σας. Αφού τελειώσει η αναβάθμιση, θα χρειαστεί να πατήσουμε το refresh του web browser ώστε ν' ανανεωθεί η σελίδα. Θα μας υποδεχτεί το νέο πάνελ διαχείρισης.



Η καρτέλα Services αποτελεί το σημαντικότερο μέρος του dd-wrt. Εδώ θα επανέλθουμε και σε επόμενο άρθρο.

κάποιος software έλεγχος ή όντως κάτι αλλάζει. Ακόμη και μια μνήμη να είναι άλλη μάρκα, θα μας δημιουργήσει σοβαρά προβλήματα.

*Ένα αρκετά ενδιαφέρον feature του dd-wrt είναι η προσθήκη *δικών* μας διαφημίσεων σε όλους τους WiFi χρήστες. Φυσικά, αυτό θα είχε νόημα περισσότερο σε κάποια καφετέρια, αλλά τι θα λέγατε να στήναμε ένα free και open WiFi access point, από το σπίτι; (Σ.τ.Ε. Αν είναι να σερβίρει διαφημίσεις του περιοδικού, είμαστε μέσα :D)*



Για αρχή, θα δώσουμε έναν κωδικό τον οποίο θα χρησιμοποιούμε για να συνδεόμαστε στον router. Καλό θα ήταν να μην τον ξεχάσουμε, αφού η ανάκτηση του είναι πολύ δύσκολη.

Το μενού είναι χωρισμένο όπως σε κάθε πάνελ διαχείρισης router. Όλα ξεκινάνε από μια σελίδα που εμφανίζει πληροφορίες για το σύστημα όπως την ώρα, τη διεύθυνση IP, τους συνδεδεμένους χρήστες κ.ο.κ. Στο μενού του dd-wrt μπορούμε να δούμε επιπλέον στατιστικά όπως διαθέσιμη μνήμη, πληροφορίες για τον επεξεργαστή και άλλα.

Στην καρτέλα Setup γίνονται όλες οι ρυθμίσεις που επηρεάζουν τη σύνδεσή μας στο Internet. Όπως θα καταλάβατε, πρόκειται για τη θύρα WAN του router κι από το Setup μπορούμε να τη διαχειριστούμε κατάλληλα. Επιπλέον, υπάρχουν ρυθμίσεις για την ώρα και το hostname του router. Δεν χρειάζεται να αλλάξουμε κάτι για αρχή, εκτός αν θέλουμε τον router μας να έχει στατική διεύθυνση IP (Σ.τ.Ε. Δεν μπορώ να φανταστώ γιατί να μην το θέλουμε, ειδικά αν μας ενδιαφέρει η απομακρυσμένη πρόσβαση στο οικιακό δίκτυο.) Επιπλέον, υπάρχουν ρυθμίσεις όπως Dynamic DNS, network bridges και άλλες.

Προχωρώντας βρίσκουμε την καρτέλα Wireless, όπου συγκεντρώνονται όλες οι ρυθμίσεις για την ασύρματη επικοινωνία του router μας. Εδώ θα χρειαστεί να ορίσουμε κρυπτογράφηση και πιστοποίηση άμεσα, εκτός αν θέλουμε να μπαίνει ο καθένας στο δίκτυό μας. Επιπλέον, θα βρούμε επιλογές για MAC authentication και WDS, αλλά ...σιγά. Αυτά τα είχαμε και με το προηγούμενο firmware :)

Για τις υπηρεσίες που βρίσκουμε στην καρτέλα Services θα συζητήσουμε σε επόμενο άρθρο.

Συνεχίζουμε στην καρτέλα Security, όπου και μας δίνεται πλήρης έλεγχος του firewall. Μπορούμε να φιλτράρουμε κάθε λογής περιεχόμενο, από browser cookies μέχρι Java applets. Αν θέλουμε κόβουμε το πρωτόκολλο ICMP, την πρόσβαση SSH και πολλά ακόμη.

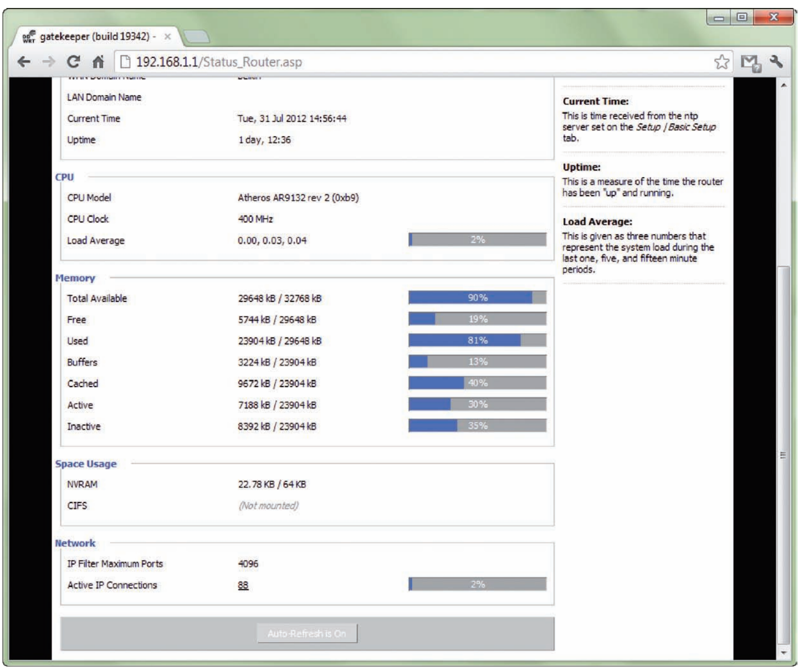
Φυσικά δεν θα μπορούσαν να λείπουν και οι βασικές επιλογές για NAT, QoS, port forwarding και DMZ. Αυτές είναι βασικές σε όλα τα routers και δεν αλλάζουν σχεδόν καθόλου από συσκευή σε συσκευή. Επιπλέον το dd-wrt προσφέρει ένα υπερπλήρες parental control με δυνατότητα να κόβει την πρόσβαση είτε σε ολόκληρο το Internet είτε σε μεμονωμένες σελίδες για κάθε συσκευή. Επιπλέον παρέχει δυνατότητα φιλτραρίσματος δεκάδων P2P δικτύων, χωρίς να επηρεάζεται καθόλου το υπόλοιπο traffic.

Τέλος, στην καρτέλα Administration βρίσκονται οι ρυθμίσεις που έχουν να κάνουν με την ίδια τη διαχείριση του router. Εδώ μπορούμε ν' αλλάξουμε τον κωδικό διαχείρισης, να επιλέξουμε πώς θέλουμε να μπαίνουμε στο πάνελ διαχείρισης, να ενεργοποιήσουμε τον ενσωματωμένο SSH/telnet server και πολλά ακόμη. Σημαντικό είναι να ενεργοποιήσουμε το Reset Button, ώστε σε περίπτωση που κάτι στραβώσει να μπορούμε εύκολα να επαναφέρουμε όλες τις προεπιλογές.

Αντί επιλόγου

Το modem/router που έχουμε από τον ISP μας, ας το παραδεχτούμε, είναι για γέλια. Γι' αυτό και ξεχνάμε το router-κομμάτι, το παίρνουμε απόφαση ότι πρόκειται για ένα απλό *DSL modem κι αγοράζουμε έναν ωραιότατο (wireless) router. Κάνει τις βασικές δουλειές αξιοπρεπέστατα και πλέον είμαστε χαρούμενοι. Του εγκαθιστούμε το dd-wrt και ξαφνικά κάνει και καμιά 100στή άλλες δουλειές, αξιόπιστα κι εντελώς δωρεάν! Πλέον, είμαστε μετά το "χαρούμενοι".

Το παρόν άρθρο δεν ήταν παρά μια εισαγωγή στον κόσμο του dd-wrt. Μην ξεχνάτε ότι πρόκειται για μια πληρέστατη διανομή Linux. Θεωρητικά, μπορούμε να τρέξουμε όποια εφαρμογή θέλουμε -- ακόμη και X server να σηκώσουμε. Αλλά γι' αυτό --και πολλά άλλα-- θα συζητήσουμε σε επόμενο άρθρο.



Η καρτέλα Status περιέχει όλες τις πληροφορίες για το σύστημα. Μπορούμε να δούμε χρήση CPU, μνήμης, διαθέσιμο αποθηκευτικό χώρο κ.ά.



Skill:
Advanced

Tags:
botnets, PHP, IRC

Δημιουργία ενός Botnet από την αρχή!

by Ion

Όπως υποσχεθήκαμε στο άρθρο της σελίδας 12, θα εξερευνήσουμε τώρα τα ενδότερα της λειτουργίας των botnets, ώστε να καταλάβουμε και πρακτικά πλέον πώς αυτά μπορούν να επηρεάσουν την ασφάλεια δικτύων και υπολογιστών. Αυτό θα γίνει φτιάχνοντας το *δικό μας* υποτυπώδες botnet, στο οποίο και θα δίνουμε εντολές από την άνεση της πολυθρόνας μας κι εκείνο θα εκτελεί. Καιρός να δούμε πώς είναι να βρίσκεται κανείς στη θέση του botmaster, τι λέτε; Προσέξτε μόνο να μην σας αρέσει πολύ :D

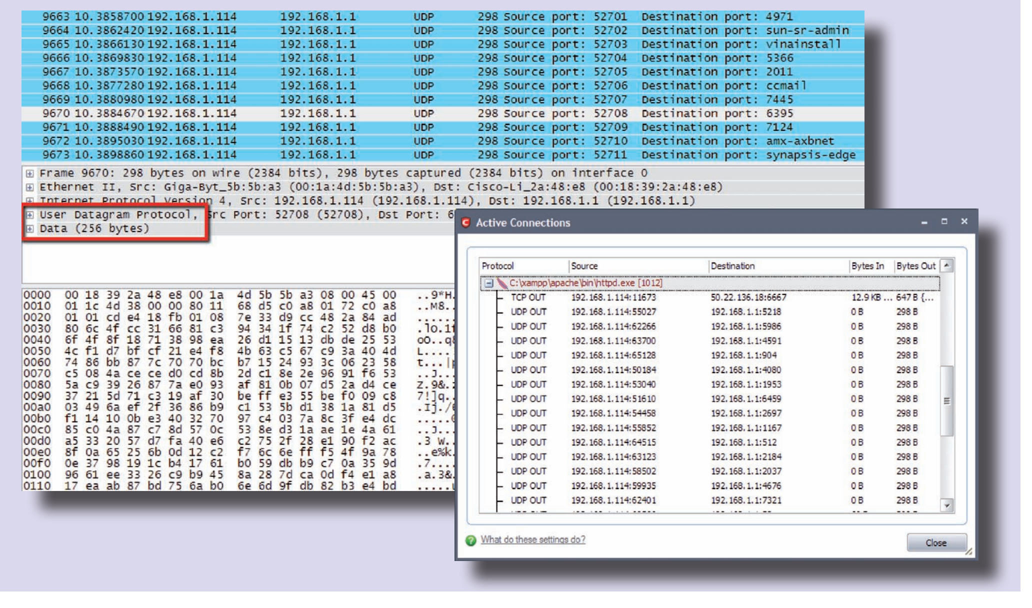
Για να φτιαχτεί ένα botnet πρέπει να έχουμε έτοιμο κάποιο πρόγραμμα, το οποίο θα μας επιτρέψει να δημιουργήσουμε ένα δίκτυο αποτελούμενο από όλους τους υπολογιστές-zombies που θα εντάξουμε σε αυτό. Μπορούμε να βρούμε κάποιο έτοιμο bot με λίγο ψάξιμο στα κατάλληλα μέρη, αλλά μπορούμε να προγραμματίσουμε και το δικό μας -- από την αρχή.

Η συγγραφή ενός bot μπορεί να γίνει μια αρκετά πολύπλοκη υπόθεση. Στα εκπαιδευτικά πλαίσια που θα λειτουργήσουμε εμείς, σίγουρα μπορούμε να είμαστε κάπως περισσότερο ανεκτικοί με την ποιότητα του τελικού προγράμματος. Δεν μειώνεται όμως η επιθυμία μας να δημιουργήσουμε κάτι χρήσιμο και αποτελεσματικό -- και φυσικά να μάθουμε μέσα από την όλη διαδικασία.

Πριν αρχίσουμε τη συγγραφή του προγράμματος πρέπει να αναλογιστούμε την αρχιτεκτονική του botnet και πώς οι διάφοροι υπολογιστές θα γίνονται zombies. Με άλλα λόγια, πώς θα καταστεί δυνατόν να εγκαθιστούμε απομακρυσμένα το bot μας σε αυτούς αλλά και να το εκτελούμε. Επιπρόσθετα, θα πρέπει να σκεφτούμε ποιος είναι ο καλύτερος τρόπος για να ελέγχουμε απομακρυσμένα όλα αυτά τα μηχανήματα.

Μετά από σκέψη καταλήγουμε στο εξής -- και πιστεύουμε θα συμφωνήσετε μαζί μας: Ο ευκολότερος τρόπος είναι να χρησιμοποιήσουμε το πρωτόκολλο IRC για το κέντρο C&C, καθώς και τη γλώσσα PHP για το bot. Το γιατί είναι απλό. Όπως είπαμε, πρέπει με κάποιον τρόπο το bot να εγκαθίσταται σε διάφορους υπολογιστές οι οποίοι και θα περνούν στον έλεγχο μας. Ένας από τους ευκολότερους τρόπους είναι να εκμεταλλευτούμε τις αδυναμίες σε διαδικτυακές εφαρμογές, όπως κάποιο CMS, ένα forum κ.λπ. Αν κάποια τέτοια εφαρμογή έχει ένα σφάλμα που επιτρέπει σε χρήστες να κάνουν upload αρχεία μη επιθυμητού τύπου, όπως PHP scripts, αυτομάτως έχουμε έναν εύκολο τρόπο στα χέρια μας για να ανεβάσουμε το bot μας στον εκάστοτε server που φιλοξενεί αυτή την ευπαθή εφαρμογή. Τέτοιες αδυναμίες δεν είναι σπάνιες και μπορείτε να τις βρείτε ως arbitrary ή unrestricted file upload vulnerabilities, στα γνωστά sites

Τεστάροντας το bot μας τοπικά, πραγματοποιούμε μια επίθεση UDP flood προς το οικιακό Linksys router. Ορίστε πώς την καταγράφει το Comodo Firewall -- και το γνωστό εργαλείο Wireshark.



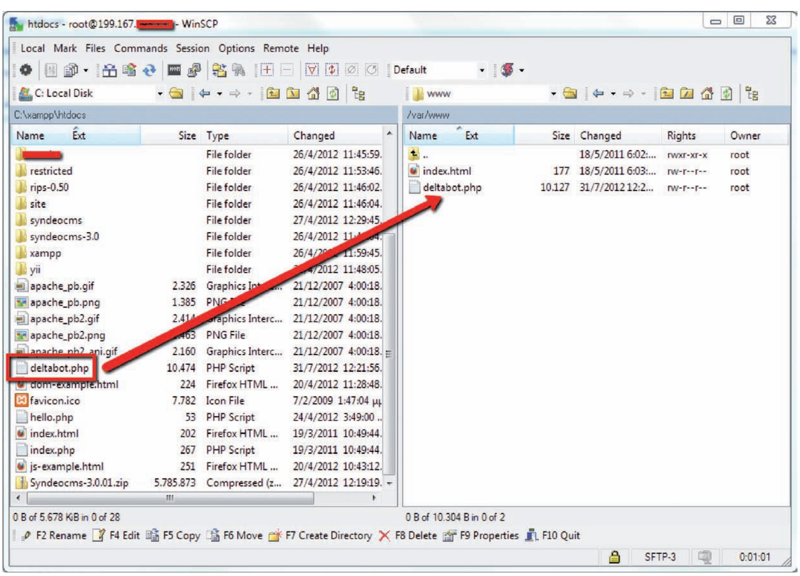
που καθημερινά συγκεντρώνουν exploits, όπως είναι για παράδειγμα το Exploit-DB (www.exploit-db.com).

Για το κέντρο C&C, το πρωτόκολλο IRC είναι αυτό που μας βολεύει. Το σκεπτικό μας είναι το ακόλουθο: Αφού ανεβάσουμε το script του bot μας σε αρκετούς servers, τότε καλώντας το αυτό θα πραγματοποιεί μια μόνιμη σύνδεση με τον κεντρικό IRC server, σε ένα συγκεκριμένο κανάλι. Ο IRC server θα επιτελεί καθήκοντα κέντρου C&C και στο κανάλι θα είμαστε και εμείς, όπου ανά πάσα στιγμή θα βλέπουμε όλα τα συνδεδεμένα zombies. Εκεί θα μπορούμε να δίνουμε τις εντολές μας και να βλέπουμε σε πραγματικό χρόνο την απόκριση των bots. Πάμε για PHP και IRC, λοιπόν!

Αφού ξεκαθαρίσαμε τις τεχνολογίες που θα χρησιμοποιήσουμε, έχουμε ένα ακόμη ένα μικρό θέμα. Πού θα βρούμε servers, για να τους κάνουμε zombies; Επειδή φυσικά δεν θέλουμε να ανακατευτούμε με ξένη περιουσία, θα καταφύγουμε στην εξής λύση: Θα επιστρατεύσουμε διάφορα VPS που έχουμε online ώστε όλα μαζί να ενταχθούν στο πείραμα του botnet μας. Αυτά θα είναι πέντε στον αριθμό και υπάρχει κι ένα ακόμα, το οποίο θα γίνει ο στόχος μας λίγο αργότερα. Ναι, ίσως να μην φαίνονται πολλά, αλλά σκεφτείτε πως με την εκμετάλλευση μίας μόνο αδυναμίας που επιτρέπει παράτυπο file upload, προσθέσαμε δοκιμαστικά άλλα εφτά zombies μέσα σε κάτι λιγότερο από μία ώρα χαλαρής ενασχόλησης. Το να μεγαλώσει επομένως το botnet είναι κάτι που απλά θέλει χρόνο και κάποιες όχι-και-τόσο εξεζητημένες γνώσεις. Αυτό όμως είναι κάτι που για την ώρα θα αφήσουμε στην ευχέρεια σας...

Το πρωτόκολλο IRC. Μιλάτε; Σας ακούμε!

Πριν περάσουμε στη συγγραφή του bot μας, θα πρέπει να μάθουμε ορισμένα πράγματα για το πρωτόκολλο που θα χρησιμοποιήσουμε για την επικοινωνία εντός botnet. Πρέπει φυσικά να γνωρίσουμε το IRC, γιατί αλλιώς δε θα είμαστε σε θέση να στέλνουμε στο κέντρο C&C μηνύματα με τη μορφή που πρέπει, ούτε ν' απαντάμε στα "εισερχόμενα".



Για ν' ανεβάσουμε το bot μας σε διάφορους servers, χρειάζεται να κάνουμε exploit ορισμένες αδυναμίες. Στη δική μας περίπτωση βέβαια αρκεί ένα εργαλείο όπως το WinSCP, ώστε να το μεταφέρουμε με ασφάλεια στα VPS που διαθέτουμε.

Το IRC (Internet Relay Chat) φτιάχτηκε τον Αύγουστο του 1988 και καταγράφηκε επίσημα το 1993. Πρόκειται δηλαδή για ένα σχετικά παλαιό πρωτόκολλο, για τα σημερινά δεδομένα. Παρόλα αυτά είναι το πιο γνωστό και ευρέως διαδεδομένο πρωτόκολλο για online chatting κι επικοινωνία πολλών ατόμων, με κείμενο και σε πραγματικό χρόνο. Για την μετάδοση δεδομένων χρησιμοποιεί το μοντέλο client-server και το πρωτόκολλο TCP. Η βασική ιδέα είναι η εξής: Υπάρχουν διάσπαρτοι κάποιοι IRC servers. Ένας χρήστης χρησιμοποιεί κάποιον IRC client για να συνδεθεί με τον IRC server, σε μια συγκεκριμένη θύρα (port). Συνήθως αυτή είναι η θύρα 6667. Κατόπιν διαλέγει ένα ψευδώνυμο (nickname) και εισέρχεται σε διάφορα κανάλια συνομιλίας (channels). Τα κανάλια συμβολίζονται με μια δίεση (#) μπροστά απ' το όνομα. Είναι σαν δωμάτια επικοινωνίας, όπου πολλοί χρήστες γράφουν στο κεντρικό παράθυρο του καναλιού και συζητούν μεταξύ τους σε πραγματικό χρόνο. Τα μηνύματά τους αυτά είναι ορατά σε όλους τους άλλους χρήστες, οι οποίοι επίσης βρίσκονται μέσα στο κανάλι. Κάποια άτομα τώρα, μέσα στο κανάλι, καλούνται operators και είναι οι διαχειριστές. Μπορούν μεταξύ άλλων να διώξουν όποιον θέλουν προσωρινά ή μόνιμα, να θέσουν πρόσβαση με κωδικό στο συγκεκριμένο κανάλι κ.ο.κ. Αυτοί έχουν μπροστά από το ψευδώνυμο τους το σύμβολο παπάκι (@).

Τα βασικά μηνύματα που θα ανταλλάσουμε με τον IRC server και θα μας απασχολήσουν κατά την ανάπτυξη του bot, είναι τα ακόλουθα.

JOIN. Τα μηνύματα JOIN χρησιμοποιούνται για την είσοδο ενός client σε κάποιο κανάλι συνομιλίας. Το όνομα του καναλιού και ενδεχομένως ο κωδικός πρόσβασης σε αυτό δίνονται όλα μαζί, στο ίδιο μήνυμα.

PART. το αντίθετο του JOIN. Προκαλεί έξοδο από ένα κανάλι συνομιλίας, στο οποίο ήδη είμαστε συνδεδεμένοι.

PRIVMSG. Το PRIVMSG, όπως υποδηλώνει και η ονομασία του, χρησιμοποιείται για να στέλνουμε προσωπικά μηνύματα. Αυτά μπορεί να έχουν παραλήπτη έναν συγκεκριμένο χρήστη ή κι ένα κανάλι. Βέβαια, στην περίπτωση του καναλιού το μήνυμά μας θα το δουν όλοι οι συνδεδεμένοι χρήστες στο κανάλι. Και πάλι όμως θεωρείται "προσωπικό", από την άποψη ότι επιλέγουμε μια συγκεκριμένη ομάδα χρηστών που θα το παραλάβει κι όχι όλους όσοι είναι συνδεδεμένοι στον IRC server. Όλη η επικοινωνία (συνομιλία) στο IRC γίνεται με μηνύματα τύπου PRIVMSG, που στέλνουν οι συνδεδεμένοι χρήστες.

PING. το PING είναι ένας ειδικός τύπος μηνύματος, ο οποίος στέλνεται από τον IRC server ανά τακτά χρονικά διαστήματα σε κάθε client για να δει αν είναι όντως συνδεδεμένος. Ο client θα πρέπει να απαντήσει με ένα μήνυμα PONG, για να υποδηλώσει ότι είναι online και παραμένει συνδεδεμένος.

USER, NICK και **PASS.** Οι τρεις αυτοί τύποι μηνυμάτων έχουν να κάνουν με τη διαχείριση της σύνδεσης του κάθε client, στον IRC server. Συγκεκριμένα, ένα μήνυμα USER στέλνεται στην αρχή της σύνδεσης για να καθορίσει το username, το hostname, το αληθινό όνομα και άλλα στοιχεία του client. Το NICK χρησιμοποιείται για να τεθεί το ψευδώνυμο που θα έχει ο client και, τέλος, το PASS για να σταλεί στον server ο ενδεχόμενος κωδικός του ψευδωνύμου (αν πρόκειται για εγγεγραμμένο χρήστη).

Μία λίστα με λεπτομέρειες για όλες τις εντολές/μηνύματα του IRC, μπορείτε να δείτε στη Wikipedia:

http://en.wikipedia.org/wiki/List_of_Internet_Relay_Chat_commands

Επίσης, συγκεκριμένη αναφορά πρέπει να γίνει στη μορφή (format) των μηνυμάτων που ανταλλάσσονται. Όλα τα μηνύματα που θα δίνουμε στον server θα λαμβάνονται από τους clients (δηλαδή τα bots), τα οποία θα πρέπει να είναι σε θέση να τα αποκωδικοποιούν!

Κατ' αρχάς, αφού τις εντολές μας θα τις δίνουμε μέσα σε ένα κανάλι ή σε κάποιο bot μεμονωμένα, αναφερόμαστε σε μηνύματα τύπου PRIVMSG. Αυτό που συμβαίνει είναι το εξής: Όταν στέλνουμε ένα μήνυμα τύπου PRIVMSG σε ένα κανάλι, ο server το λαμβάνει και το στέλνει σε όλους τους χρήστες που είναι συνδεδεμένοι στο ίδιο κανάλι. Έτσι εμφανίζεται και στο κεντρικό παράθυρο συνομιλίας που βλέπει ο κάθε χρήστης. Η μορφή των PRIVMSG μηνυμάτων που στέλνουν οι clients στον server και η μορφή των PRIVMSG μηνυμάτων που παίρνουν οι clients από τον server, δεν είναι η ίδια -- ακόμα και αν μιλάμε για το ίδιο μήνυμα που μας επιστρέφεται. Εμείς πρέπει να μάθουμε και τις δύο μορφές. Τη μία για να μπορούμε να στέλνουμε σωστά μηνύματα στον server και τη δεύτερη για να καταλαβαίνουμε αυτά που λαμβάνουμε από τον server. Η μορφή ενός PRIVMSG που λαμβάνουμε από τον server είναι η ακόλουθη:

```
:Ψευδώνυμο!Όνομα@Host PRIVMSG #Κανάλι :Μήνυμα
```

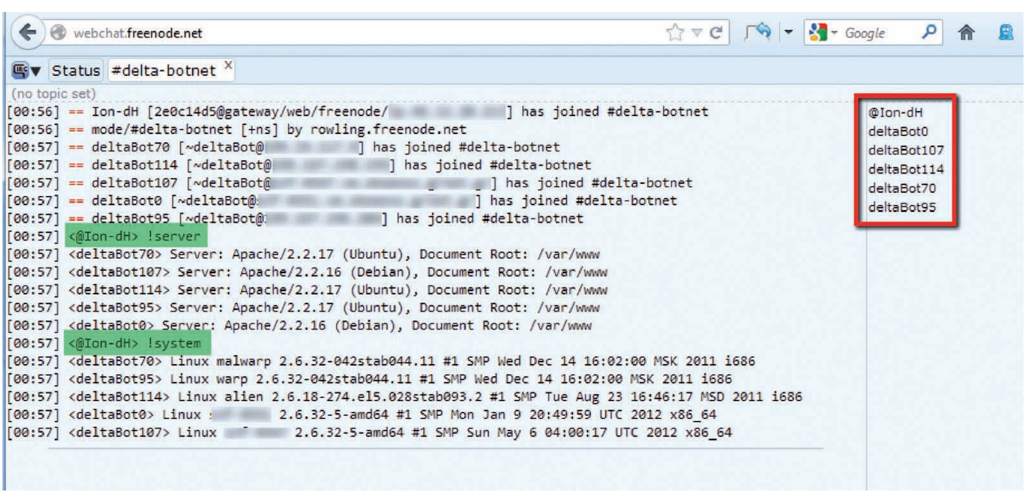
Λέει, απλά, ότι ο χρήστης με το ψευδώνυμο τάδε και όνομα δείνα, έστειλε ένα PRIVMSG στο συγκεκριμένο κανάλι το οποίο περιείχε το ακόλουθο κείμενο. Τα σημεία στίξης που χρησιμοποιούνται παίζουν διαχωριστικό ρόλο μεταξύ των διαφορετικών οντοτήτων.

Όταν τώρα στέλνουμε κάτι στον server, το κομμάτι πριν τη λέξη PRIVMSG εισέρχεται αυτόματα και δε χρειάζεται να το βάζουμε χειροκίνητα. Άρα, η μορφή ενός μηνύματος PRIVMSG, το οποίο θέλουμε εμείς να στείλουμε θα έχει τη μορφή:

```
PRIVMSG #Κανάλι :Μήνυμα
```

Διεφάνη εξάλλου ότι αραιά και που ο server θα μας στέλνει μηνύματα PING. Αυτά έχουν την παρακάτω μορφή:

```
PING :Μήνυμα
```



Η εικόνα του καναλιού συνομιλίας του botnet μας, με όλες τις μονάδες σε στάση προσοχής! Εδώ βλέπουμε το αποτέλεσμα της εκτέλεσης των εντολών !server και !system.

Στα μηνύματα PING θα πρέπει να απαντάμε με ένα μήνυμα PONG, της μορφής:

```
PONG :Μήνυμα
```

Όπου «:Μήνυμα» είναι το ίδιο αλφαριθμητικό που λάβαμε προηγουμένως με το PING.

Τέλος, πρέπει να θυμόμαστε ότι κάθε τύπος μηνύματος από και προς τον server, τελειώνει με τους χαρακτήρες \r\n που στην ουσία υποδηλώνουν το τέλος του μηνύματος και την αλλαγή γραμμής στο κανάλι συνομιλίας. Όλα αυτά είναι σημαντικά για το πώς το bot μας θα μπορεί να δέχεται τις εντολές που του δίνουμε, αλλά και να αποκρίνεται ώστε να μας ενημερώνει για την επιτυχία ή την αποτυχία τους. Αν όλα αυτά σας μπέρδεψαν, μην σας απασχολούν πολύ για την ώρα. Θα τα συζητήσουμε ξανά σε λίγο, καθώς θα εξετάζουμε τον κώδικα του bot βήμα προς βήμα.

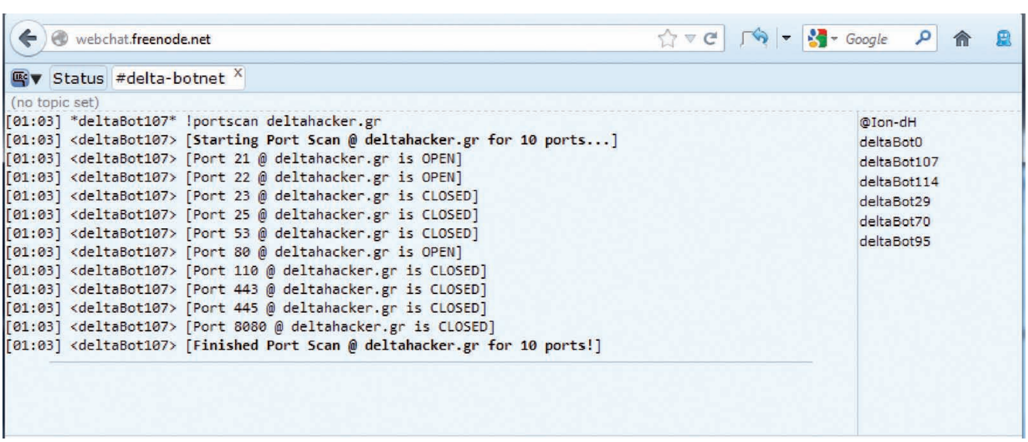
deltaBots!

Το πρώτο στάδιο για την κατασκευή του bot μας είναι η σχεδίαση του σε υψηλό, όπως λέμε, επίπεδο. Με απλά λόγια, χρειάζεται να σκεφτούμε τι ακριβώς θα θέλαμε να κάνει, ώστε να βρούμε τον κατάλληλο τρόπο για το πώς ακριβώς θα το προγραμματίσουμε. Εμείς θα εργαστούμε ως εξής: Θα φτιάξουμε μια κλάση (class) που αντιπροσωπεύει το bot, η οποία θα περιέχει διάφορα χαρακτηριστικά και χρήσιμες συναρτήσεις. Τέτοια χαρακτηριστικά θα είναι, π.χ., το ψευδώνυμο που θα χρησιμοποιεί και η διεύθυνση του IRC server στον οποίο θα συνδέεται, ενώ θα υπάρχουν συναρτήσεις για τις διάφορες ενέργειες που θα πραγματοποιεί όπως αποστολή μηνύματος στον server, αποχώρηση από κάποιο κανάλι κ.ά. Μένει επομένως να ορίσουμε ποια θα είναι αυτά τα χαρακτηριστικά και τι ενέργειες θα θέλαμε να πραγματοποιεί (μια διαδικασία που στην ανάπτυξη εφαρμογών ονομάζεται "καθορισμός απαιτήσεων"). Εμείς φυσικά δεν θα είμαστε αυστηροί σε όλα αυτά, αλλά σε γενικές γραμμές θα κάνουμε τις επιλογές μας οι οποίες κατά τη συγγραφή του bot ενδέχεται ν' αλλάζουν.

Κατ' αρχάς, δίνουμε ονόματα σε bot και botnet. Εδώ μπορείτε να βάλετε τη φαντασία σας να δουλέψει. Εμείς θα διαλέξουμε ως όνομα για κάποιο bot τη λέξη deltaBot ακολουθούμενη από έναν τυχαίο αριθμό, ώστε να ξεχωρίζουν μεταξύ τους. Για το botnet προτιμούμε το delta-botnet.

Αποφασίζουμε ότι το bot μας θα συνδέεται σε έναν από τους γνωστούς

Το deltaBot107 σε δράση, μόλις πραγματοποίησε ένα γρήγορο σκανάρισμα για ανοικτές θύρες στο deltahacker.gr!



IRC servers, το δίκτυο Freenode, στη θύρα 6667. Το κανάλι στο οποίο θα συνδέεται θα έχει φυσικά το ίδιο όνομα με το botnet μας.

Σε γενικές γραμμές, αντιλαμβανόμαστε ότι θα πρέπει να γράψουμε τις ακόλουθες συναρτήσεις:

- Μια συνάρτηση **start()**, η οποία θα είναι υπεύθυνη για το ξεκίνημα της λειτουργίας του bot και την αρχική σύνδεση με τον IRC server.
- Μια συνάρτηση **send()**, που θα αναλαμβάνει την αποστολή μηνυμάτων προς τον IRC server.
- Συναρτήσεις **join()** και **part()**, για την είσοδο κι έξοδο από κανάλια συνομιλίας.
- Συναρτήσεις **restart()** και **shutdown()**, για τη διαχείριση του bot.
- Μια συνάρτηση **help()**, για την εμφάνιση των διαθέσιμων επιλογών.

Αυτές είναι που θα αποτελέσουν τον σκελετό του προγράμματος. Από εκεί κι έπειτα θα πρέπει να προσθέσουμε βοηθητικές συναρτήσεις --όπου χρειάζονται-- καθώς και συναρτήσεις για τις προχωρημένες λειτουργίες του bot. Εμείς κάναμε μια λίστα με λειτουργίες τις οποίες θα θέλαμε να επιτελεί και όσες περισσότερες από αυτές ήταν δυνατόν να υλοποιηθούν βρίσκονται ήδη ενσωματωμένες σ' αυτό. Έτσι, στην παρούσα έκδοση (0.1) του deltaBot, εντάξαμε τις ακόλουθες δυνατότητες.

- Εμφάνιση του uptime για κάθε zombie στο οποίο έχουμε εγκαταστήσει το bot, δηλαδή το χρονικό διάστημα που βρίσκεται σε λειτουργία.
- Εμφάνιση πληροφοριών για την έκδοση του web server που τρέχει σε κάθε zombie και εκτελεί το bot μας.
- Εμφάνιση πληροφοριών του συστήματος κάθε απομακρυσμένου zombie.
- Εμφάνιση της τρέχουσας έκδοσης του ίδιου του deltaBot.
- Αποστολή μεμονωμένου PRIVMSG σε κάποιο χρήστη ή συγκεκριμένο κανάλι συνομιλίας.

Οι παραπάνω αποτελούν σχετικά απλές λειτουργίες που απώτερο σκοπό έχουν την ανταλλαγή ενημερωτικών πληροφοριών και την εξαγωγή κάποιων στατιστικών για το botnet μας. Φυσικά, θα πρέπει να προσθέσουμε και μερικές πιο εξεζητημένες δυνατότητες, όπως οι ακόλουθες:

- **Λήψη** ενός αρχείου από πλευράς bot και αποθήκευση αυτού τοπικά, στο zombie.
- **Τσουνάμι**, δηλαδή μια απλή επίθεση IRC, όπου όλα τα bots στέλνουν μαζικά μηνύματα σε κάποιον χρήστη ή κανάλι, προκαλώντας συμφόρηση.
- **Port scan**, δηλαδή σάρωση ενός host για ανοικτές θύρες (ports).
- Επίθεση **UDP flood**, όπου στέλνουμε πακέτα UDP σ' έναν host με σκοπό να προκαλέσουμε denial of service!

Οι παραπάνω λειτουργίες χρησιμοποιούν ανάλογες βοηθητικές συναρτήσεις, που αναλαμβάνουν τις λεπτομέρειες της εκτέλεσης. Αυτή η λίστα αλλάζει διαρκώς κι εννοείται πως έχετε το ελεύθερο να υλοποιήσετε τις

δικές σας μεθόδους -- και φυσικά να τις προσθέσετε στο πρόγραμμα. (Σ.τ.Ε. «Αισθανθείτε ελεύθεροι» να την τροποποιήσετε, για να μιλήσουμε σωστά Ελληνικά.) Μερικές ιδέες για πιθανές επεκτάσεις θα δώσουμε στο τέλος του άρθρου.

I'm a machine!

Τον πηγαίο κώδικα του deltaBot σε PHP μπορείτε να τον κατεβάσετε από το ακόλουθο link:

http://bit.ly/dh011deltabot

Υπενθυμίζεται ότι μπορείτε να χρησιμοποιήσετε το γνωστό πακέτο λογισμικού XAMPP σε περιβάλλον Windows, ώστε να τεστάρετε άμεσα τη λειτουργικότητα του. Αρκεί να συνδεθείτε πρώτα στον κατάλληλο IRC server και στο κανάλι συνομιλίας που ορίσαμε. Ας εξετάσουμε τον κώδικα του bot, ώστε να εξηγήσουμε τι κάνει κάθε block εντολών. (Για την ευκολότερη παρακολούθηση της ανάλυσης που ακολουθεί, ανοίξτε το deltabot.php με έναν editor που υποστηρίζει syntax highlighting αλλά και αρίθμηση γραμμών.)

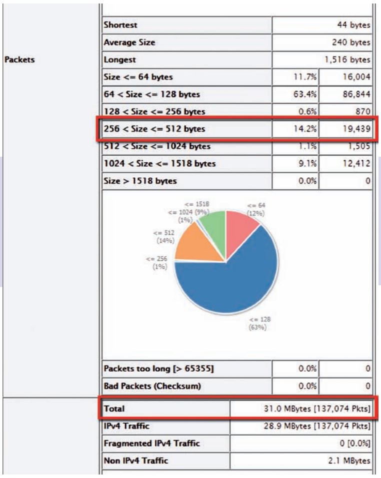
Γραμμές 27-28. Εδώ δίνουμε κάποιες οδηγίες σχετικά με την εκτέλεση του script μας. Συγκεκριμένα, επειδή η εκτέλεση ενός PHP script δεν μπορεί να διαρκεί αιωνίως και συνήθως σταματάει απ' τον ίδιο τον server, πρέπει να δηλώσουμε συγκεκριμένα ότι δεν θέλουμε να συμβεί αυτό. Γι' αυτό και βάζουμε χρονικό όριο το μηδέν, που σημαίνει ότι το bot μας δε θα σταματήσει αν δεν το θελήσουμε εμείς. Δεύτερον, ενεργοποιούμε την εμφάνιση σφαλμάτων στον browser κατά την εκτέλεση του bot, ώστε να ενημερωθούμε αν για κάποιον λόγο κάτι δεν λειτουργήσει σωστά.

Γραμμή 30. Δηλώνουμε την κλάση του deltaBot, μέσα στην οποία έχουμε γράψει οτιδήποτε το αφορά. Για όσους δεν ασχολείστε με τον αντικειμενοστρεφή προγραμματισμό, μια κλάση είναι στην ουσία μια λογική οντότητα που αντιπροσωπεύει κάτι -- στην περίπτωση μας το deltaBot.

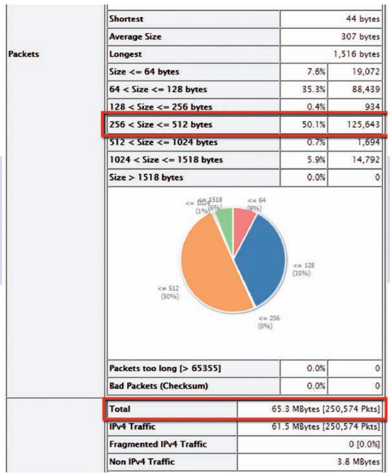
Γραμμή 33. Αυτή είναι μια απλή μεταβλητή που παίρνει τιμή TRUE ή FALSE από εμάς, αναλόγως αν τρέχουμε το bot τοπικά στον υπολογιστή μας για δοκιμή ή αν όντως έχουμε ανεβάσει το script σε κάποιο zombie. Χρησιμοποιείται παρακάτω κατά την αποστολή του μηνύματος USER στον IRC server, ώστε να σταλούν σωστά στοιχεία που αφορούν στην IP, στο hostname κ.λπ. του bot.

Γραμμή 36. Μια μεταβλητή που αντιστοιχεί στην τρέχουσα έκδοση του deltaBot.

Γραμμές 39-47. Εδώ έχουμε φτιάξει έναν πίνακα (array) με το όνομα config, που περιέχει όλα τα στοιχεία σχετικά με τη σύνδεση του bot με τον IRC server. Το καλό με τα arrays στην PHP είναι ότι μπορούν κι αποθηκεύουν ζευγάρια δεδομένων, αυτό που ονομάζουμε key-value pairs. Αυτό σημαίνει πως αν προσπελάσουμε τον πίνακα στη θέση key, τότε επιστρέφεται το αντίστοιχο value με το οποίο υπάρχει συσχέτιση μέσω του τελεστή =>. Εδώ έχουμε κάνει τις παρακάτω επιλογές: Το bot μας θα συνδέεται στον IRC server Freenode, στη θύρα 6667. Θα εισέρχεται στο κανάλι με όνομα #delta-botnet και θα έχει ως όνομα (όχι ψευδώνυμο) την περιγραφή που έχουμε επιλέξει. Το ψευδώνυμό



Packets	Shortest		44 bytes
	Average Size		240 bytes
	Longest		1,516 bytes
	Size <= 64 bytes	11.7%	16,004
	64 < Size <= 128 bytes	63.4%	86,844
	128 < Size <= 256 bytes	0.6%	870
	256 < Size <= 512 bytes	14.2%	19,439
	512 < Size <= 1024 bytes	1.1%	1,505
	1024 < Size <= 1518 bytes	9.1%	12,412
	Size > 1518 bytes	0.0%	0
	Packets too long [> 65355]	0.0%	0
	Bad Packets (Checksum)	0.0%	0
	Total		31.0 MBytes [137,074 Pkts]
	IPv4 Traffic		28.9 MBytes [137,074 Pkts]
	Fragmented IPv4 Traffic		0 [0.0%]
	Non IPv4 Traffic		2.1 MBytes



Packets	Shortest		44 bytes
	Average Size		307 bytes
	Longest		1,516 bytes
	Size <= 64 bytes	7.6%	19,072
	64 < Size <= 128 bytes	35.3%	88,439
	128 < Size <= 256 bytes	0.4%	934
	256 < Size <= 512 bytes	50.1%	125,643
	512 < Size <= 1024 bytes	0.7%	1,694
	1024 < Size <= 1518 bytes	5.9%	14,792
	Size > 1518 bytes	0.0%	0
	Packets too long [> 65355]	0.0%	0
	Bad Packets (Checksum)	0.0%	0
	Total		65.3 MBytes [250,574 Pkts]
	IPv4 Traffic		61.5 MBytes [250,574 Pkts]
	Fragmented IPv4 Traffic		0 [0.0%]
	Non IPv4 Traffic		3.8 MBytes

Πραγματοποιώντας μια μικρής διάρκειας UDP flood attack, στο VPS μας. Παρατηρείστε πόσο αλλάζει ο αριθμός των πακέτων που δέχεται το σύστημα! Προσέξτε επίσης και τη χαρακτηριστική "πίτα" στη μέση, που δείχνει την κατανομή των πακέτων ανά μέγεθος (τα δικά μας έχουν μέγεθος 298 bytes κι αντιστοιχούν στο πορτοκαλί).

του, δηλαδή το πώς θα φαίνεται στη λίστα χρηστών, θα αποτελείται από τη λέξη deltaBot και θα ακολουθείται από ένα τυχαίο νούμερο μεταξύ του μηδέν και της τιμής maxrand, ώστε τα bots να ξεχωρίζουν μεταξύ τους (δεν επιτρέπεται δύο ή παραπάνω χρήστες να έχουν ταυτόχρονα το ίδιο ψευδώνυμο). Η προσθήκη αυτού του αριθμού γίνεται παρακάτω, στη συνάρτηση start(). Τέλος, η τιμή pass είναι προαιρετική και θα χρησιμοποιηθεί σε περίπτωση που για κάποιο λόγο θελήσετε να κατοχυρώσετε το ψευδώνυμο του bot (αφαιρώντας βέβαια τον τυχαίο αριθμό που προστίθεται στο τέλος, ώστε να είναι κάτι σταθερό).

Γραμμή 50. Αυτή είναι μια μεταβλητή στην οποία αποθηκεύουμε τη σύνδεση που κάνουμε με τον IRC server. Χρησιμοποιείται για να πάρουμε δεδομένα απ' τον IRC server και να στείλουμε μηνύματα σε αυτόν.

Γραμμή 53. Στον πίνακα ex θα αποθηκεύουμε σειραϊκά τα επιμέρους κομμάτια κάθε μηνύματος που δεχόμαστε από τον IRC server. Πρέπει, όπως είπαμε, να σπάμε κάθε μήνυμα, ώστε να το διαχωρίζουμε στα συστατικά του. Αυτά λοιπόν θα αποθηκεύονται εδώ -- για κάθε μήνυμα.

Γραμμές 56-72. Η συνάρτηση start() είναι η πρώτη και μοναδική που καλούμε χειροκίνητα, αφού δημιουργήσουμε ένα στιγμιότυπο του deltaBot. Η δουλειά της είναι πολύ απλή. Πραγματοποιεί την αρχική σύνδεση με τον IRC server μέσω της fsockopen(), κατόπιν αποστέλλει τα κατάλληλα μηνύματα USER, NICK, και μετά κάνει join στο κανάλι που ορίσαμε παραπάνω, στον πίνακα config. Τέλος, καλεί τη συνάρτηση main() που ακολουθεί, και είναι η βασική συνάρτηση του bot.

Γραμμές 75-249. Η συνάρτηση main(). Πρόκειται για τη βάση του deltaBot και το σημείο όπου πραγματοποιούνται όλες οι επιμέρους λειτουργίες του. Ας τη δούμε με μεγαλύτερη λεπτομέρεια.

Γραμμή 77. Εδώ δηλώνουμε τη βασική δομή επανάληψης, μέσα στην οποία τοποθετούνται όλες οι υπόλοιπες εντολές. Αυτό που λέει στην ουσία το συγκεκριμένο while στο deltaBot, είναι: Όσο δεν έχει κλείσει η σύνδεση με τον IRC server, εσύ να εκτελείς ό,τι υπάρχει παρακάτω.

Γραμμές 80-83. Πρώτη κίνηση είναι να πάρουμε από τον IRC server τα δεδομένα που έχει για εμάς. Συγκεκριμένα, λαμβάνουμε δεδομένα που αποθηκεύουμε στη μεταβλητή data και τα οποία εμφανίζουμε στον browser όπου εκτελείται το bot. Σημαντικό ρόλο εδώ παίζει η μέθοδος trim() της PHP, η οποία είναι απαραίτητη για να αφαιρεθούν οι χαρακτήρες \r\n οι οποίοι, όπως είπαμε, εισάγονται αυτόματα στο τέλος κάθε μηνύματος. Αυτό είναι σημαντικό για τον εξής λόγο: Όταν παρακάτω θα προσπαθήσουμε να δούμε με ποια εντολή του deltaBot ταιριάζει η εντολή που δώσαμε στο κανάλι συνομιλίας, έχει διαφορά αν η εντολή αυτή θα έχει τη μορφή "command" ή "command\r\n". Αν έχει τη δεύτερη μορφή η σύγκριση θα αποτύχει, με αποτέλεσμα να μην εκτελεστεί η ανάλογη μέθοδος/ρουτίνα.

Γραμμή 86. Όπως είπαμε, κάθε μήνυμα αποτελείται από επιμέρους στοιχεία τα οποία θα αποθηκεύουμε στον πίνακα ex. Εδώ πραγματοποιείται το σπάσιμο των μηνυμάτων με τη μέθοδο explode() της PHP. Αυτό που κάνει είναι να χωρίζει σε επιμέρους κομμάτια το μεγάλο, αυτόνομο string που περιέχει το μήνυμα, οπουδήποτε βρίσκει ένα κενό. Παράδειγμα: Για ένα μήνυμα της μορφής

```
:Ψευδώνυμο!Όνομα@Host PRIVMSG #Κανάλι :Μήνυμα
```

Το κομμάτι από την αρχική άνω-κάτω τελεία μέχρι και το Host θα έμπαινε στην πρώτη θέση του πίνακα, η λέξη PRIVMSG στη δεύτερη θέση, το #Κανάλι στην τρίτη και το :Μήνυμα στην τέταρτη.

Γραμμές 89-92. Ξεμπερδεύουμε με την απλή περίπτωση να μας έχει αποσταλεί ένα μήνυμα PING. Απαντάμε λοιπόν με ένα ανάλογο μήνυμα PONG.

Γραμμή 95. Περνάμε σιγά σιγά στο ζουμί... Εδώ παίρνουμε το τέταρτο συστατικό κάθε μηνύματος που σπάσαμε παραπάνω και το αποθηκεύουμε στη μεταβλητή command. Στην ουσία πρόκειται για την πρώτη λέξη που πληκτρολογεί κάποιος στο κανάλι συνομιλίας και αντιστοιχεί στην εντολή που δίνουμε στα bots (π.χ., !version). Είναι ex[3] κι όχι ex[4], για το λόγο ότι ξεκινάμε την αρίθμηση των θέσεων του πίνακα από το 0 και όχι από το 1.

Γραμμή 98. Εδώ έχουμε γράψει μια μεγάλη δομή επιλογής, χρησιμοποιώντας την εντολή switch. Αυτό που γίνεται σε αυτό το σημείο είναι η αντιστοίχιση της εντολής command που αποθηκεύσαμε ακριβώς παραπάνω, σε μία από τις εντολές που καταλαβαίνει το deltaBot. Όταν συμβεί κάποιο τέτοιο ταίριασμα, τότε τρέχει το κομμάτι του κώδικα που αντιστοιχεί στη συγκεκριμένη εντολή, αλλιώς απλά δε γίνεται τίποτα και ξεκινάμε πάλι απ' την αρχή παίρνοντας από τον IRC server το επόμενο μήνυμα. Πάμε να δούμε αυτές τις εντολές.

Γραμμές 102-115. Είσοδος και αποχώρηση από κανάλια συνομιλίας. Δίνονται από τον botmaster ως !join #channel και !part #channel. Μπορούμε να δώσουμε πολλά κανάλια στη σειρά, με ενδιάμεσα κενά.

Γραμμές 118-125. Εντολή που λέει στο bot να στείλει ένα συγκεκριμένο μήνυμα σε κάποιον χρήστη ή κανάλι. Δίνεται ως: !say user/#channel message. Αν προσέξετε, ανακατασκευάζει το message ενώνοντας τα σπασμένα κομμάτια του από τον πίνακα ex.

Γραμμές 128-148. Πρόκειται για τέσσερις εντολές με τις οποίες τα bots εμφανίζουν στο κεντρικό κανάλι #delta-botnet διάφορες πληρο-

φορίες, όπως: η τρέχουσα έκδοση του script του bot, το uptime κάθε zombie, το είδος και η έκδοση κάθε web server που εκτελεί το bot και πληροφορίες συστήματος για κάθε zombie (λειτουργικό σύστημα, έκδοση πυρήνα κ.λπ.). Οι εντολές δίνονται χωρίς παραμέτρους και είναι οι !version, !uptime, !server και !system.

Γραμμές 151-166. Η αναγνώριση της εντολής για τη λήψη ενός αρχείου. Δίνεται ως !download URL path. Αν δεν δοθεί κάποιο συγκεκριμένο, τοπικό path για την αποθήκευση του αρχείου σε κάθε zombie, τότε το αρχείο καταλήγει αυτόματα στη διαδρομή /tmp/file.delta. Η λήψη πραγματοποιείται μέσω της συνάρτησης download() που θα δούμε παρακάτω.

Γραμμές 169-191. Εδώ έχουμε την IRC “επίθεση” τύπου τσουνάμι. Ο botmaster δίνει την εντολή !tsunami user/#channel rounds και όλα τα bots στέλνουν το μήνυμα “TSUNAMIIII!! Hahah!!” στον επιλεγμένο χρήστη ή κανάλι συνομιλίας για rounds στο πλήθος επαναλήψεις. Αν δεν δοθεί συγκεκριμένη τιμή στο rounds, τότε πραγματοποιείται για 20 φορές. Ένα σημείο προσοχής είναι τα “\002”, όπου υπάρχουν στα μηνύματα που εμφανίζουν τα bots στο κεντρικό κανάλι του botnet. Πρόκειται για οκταδικές τιμές κωδικών που αλλάζουν το formatting ή τα χρώματα. Το \002, για παράδειγμα, αντιστοιχεί στην έντονη γραφή (bold).

Γραμμές 194-210. Το deltaBot μπορεί να πραγματοποιεί port scanning, για ν’ ανακαλύπτει αν ορισμένες θύρες είναι ανοικτές ή κλειστές. Η εντολή δίνεται ως: !portscan host/IP λίστα-θυρών-χωρισμένες-με-κόμμα. Αν δεν δοθούν συγκεκριμένες θύρες τότε σκανάρονται δέκα γνωστές θύρες κοινών υπηρεσιών όπως FTP, SSH, HTTP κλπ. Το port scanning πραγματοποιείται μέσω της συνάρτησης portscan() που θα δούμε παρακάτω.

Γραμμές 213-229. Η επίθεση τύπου UDP flood. Αυτό που κάνει είναι να δώσει εντολή σε όλα τα bots να αρχίσουν την αποστολή UDP πακέτων σε έναν server, για ορισμένο χρονικό διάστημα. Τα πακέτα δεν κατευθύνονται προς μια συγκεκριμένη θύρα αλλά σε τυχαίες. Ο στόχος είναι να δημιουργηθεί μεγάλος φόρτος εργασίας από πλευράς server, καθώς κάθε πακέτο UDP πρέπει να επεξεργαστεί από αυτόν. Συγκεκριμένα, όταν φτάνει ένα πακέτο ο υπολογιστής πρέπει να δει σε ποια θύρα κατέφτασε και να ελέγξει ποια υπηρεσία ακούει σε αυτή τη θύρα. Μόλις γίνει φανερό ότι δεν υπάρχει εφαρμογή που χρησιμοποιεί αυτή τη θύρα, απαντάει με ένα πακέτο ICMP που υποδηλώνει ότι δεν ήταν δυνατή η παράδοση του UDP πακέτου στον προορισμό του. Η αποστολή συνεχόμενων πακέτων μαζικά σε διάφορες θύρες έχει το αποτέλεσμα να γίνει ο υπολογιστής μη προσπελάσιμος διαδικτυακά. Η επίθεση UDP flood είναι γενικά εύκολα πραγματοποιήσιμη. Με τον μικρό αριθμό bots που είχαμε στη διάθεση μας καταφέραμε αρκετά ενδιαφέροντα πράγματα. Η εντολή δίνεται ως: !udpflood host/IP duration. Αν δεν δοθεί η παράμετρος duration, τότε η επίθεση πραγματοποιείται για 30 δευτερόλεπτα μέσω της συνάρτησης udpflood(), που θα δούμε παρακάτω.

Γραμμές 232-244. Άλλες τρεις λειτουργίες που αφορούν στην επανεκκίνηση του bot με την !restart, το σταμάτημά του με την !shutdown και την εμφάνιση βοήθειας, δίνοντας !help. Και οι τρεις πραγματοποιούνται με την κλήση των αντίστοιχων συναρτήσεων.

Γραμμές 252-256. Η συνάρτηση send(). Χρησιμοποιείται κατά κόρον και κάθε φορά που θέλουμε να αποστείλουμε οτιδήποτε στον IRC server. Προσέξτε ότι, σύμφωνα με τις προδιαγραφές του πρωτοκόλλου IRC, προσθέτει αυτόματα τους ειδικούς χαρακτήρες \r\n στο τέλος κάθε μηνύματος. Επίσης, εμφανίζει τα μηνύματα με έντονη γραφή στον browser όπου εκτελείται κάθε bot.

Γραμμές 258-280. Οι συναρτήσεις join(), part(), restart() και shutdown(), οι οποίες εκτελούν τις αντίστοιχες λειτουργίες για το bot μας μέσω μηνυμάτων που στέλνουν στον IRC server με τη send().

Γραμμές 283-306. Η συνάρτηση download(), που αναλαμβάνει τη λήψη του αρχείου της επιλογής μας σε κάθε zombie server. Ελέγχει κατ' αρχάς αν μπορεί να γράψει στο επιλεγμένο path. Τα δικαιώματα είναι συνήθως του χρήστη www-data --που πρόκειται για τον Apache server, ο οποίος και εκτελεί το deltaBot-- και γενικά είναι κάπως περιορισμένα. Γι' αυτό και έχουμε βάλει ως προεπιλογή τον κατάλογο /tmp, όπου γενικά όλοι οι χρήστες έχουν δικαιώματα εγγραφής. Κατά δεύτερον, ελέγχει αν μπορεί να διαβάσει το αρχείο από το link που δώσαμε και ενημερώνει αναλόγως αν αποτύχει.

Γραμμές 309-329. Η συνάρτηση portscan(), η οποία ελέγχει τη λίστα των ports που δέχεται ως ορίσματα για το αν είναι ανοικτές ή όχι. Για να το κάνει αυτό πραγματοποιεί μια σύνδεση προς κάθε μία απ' αυτές. Αν η σύνδεση είναι επιτυχής, σημαίνει πως η αντίστοιχη θύρα είναι ανοικτή. Για κάθε μία από τις θύρες που δοκιμάζει ενημερώνει καταλλήλως στο κεντρικό κανάλι IRC.

Γραμμές 332-352. Η συνάρτηση udpflood() καλείται από το αντίστοιχο block της switch μέσα στην main(), και είναι υπεύθυνη για την πραγματοποίηση της επίθεσης UDP flood. Αρχικά κατασκευάζει ένα πακέτο με τυχαία δεδομένα μεγέθους 256 bytes (συν τα headers που μπαίνουν αυτόματα) και κατόπιν το αποστέλλει συνεχώς --για όσο χρόνο διαρκεί η επίθεση-- προς μία τυχαία κάθε φορά θύρα του υπολογιστή-στόχου, μεταξύ των 0 και 8080.

Γραμμές 355-361. Η βοηθητική συνάρτηση help(). Εμφανίζει μία λίστα στο κεντρικό κανάλι με όλες τις διαθέσιμες εντολές που μπορεί να δεχτεί το bot.

Γραμμές 366-367. Μπορεί να είναι οι τελευταίες γραμμές του προγράμματος, αλλά εδώ γίνεται στην ουσία η κατασκευή του ίδιου του deltaBot φτιάχνοντας ένα στιγμιότυπο, όπως λέγεται, της κλάσης που γράψαμε παραπάνω. Ακολούθως καλείται η μέθοδος start(), ώστε να πραγματοποιηθεί η σύνδεση με τον IRC server και να αρχίσει η λειτουργία του.

Επιθέσεις UDP flood και Denial of Service!

Μετά από όλα αυτά είναι καιρός να δοκιμάσουμε την ισχύ του botnet μας στην πράξη. Συγκεκριμένα θα εκτελέσουμε μαζικές ή μεμονωμένες επιθέσεις UDP ενάντια σε έναν κακόμοιρο server, ο οποίος υπομονετικά θα δεχτεί τα πακέτα μας. Έτσι, θα δούμε εκ των έσω πώς μια μικρής κλίμακας DoS ή DDoS attack μπορεί να επηρεάσει τη λειτουργία του...

Έχουμε στη διάθεση μας πέντε διαφορετικά VPS για να χρησιμοποιήσουμε στις επιθέσεις μας. Ξεκινάμε με έναν επιτιθέμενο και διάφορες χρονι-

κές διάρκειες επιθέσεων, μετρώντας παράλληλα --με συγκεκριμένα βοηθητικά εργαλεία στον server-στόχο-- την απόδοση του δικτύου. Τέτοια είναι για παράδειγμα τα ifstat, iftop και το πολύ καλό ntop. Συνεχίζουμε προσθέτοντας κι άλλα από τα υπάρχοντα VPS στην ομάδα των επιτιθέμενων κι αυξάνοντας τη χρονική διάρκεια της επίθεσης, καταγράφοντας κάθε φορά τις αλλαγές. Πραγματοποιώντας λοιπόν διάφορες τέτοιες επιθέσεις για τυχαία χρονικά διαστήματα και με κυμαινόμενο αριθμό επιτιθέμενων VPS, παρατηρούμε ότι δημιουργείται ένας ικανοποιητικός φόρτος στο δίκτυο και τα πακέτα καταφθάνουν σωρηδόν. Κάποια ενδεικτικά αποτελέσματα μπορείτε να δείτε στις σχετικές εικόνες. Η τελική δοκιμασία; Μια μαζική επίθεση UDP flood, που θα ξεκινάει από τα πέντε VPS μας με σκοπό να θέσουμε εκτός λειτουργίας τον server-στόχο. Καθόμαστε στη θέση του botmaster, λοιπόν, και δίνουμε την εντολή στο κανάλι μας:

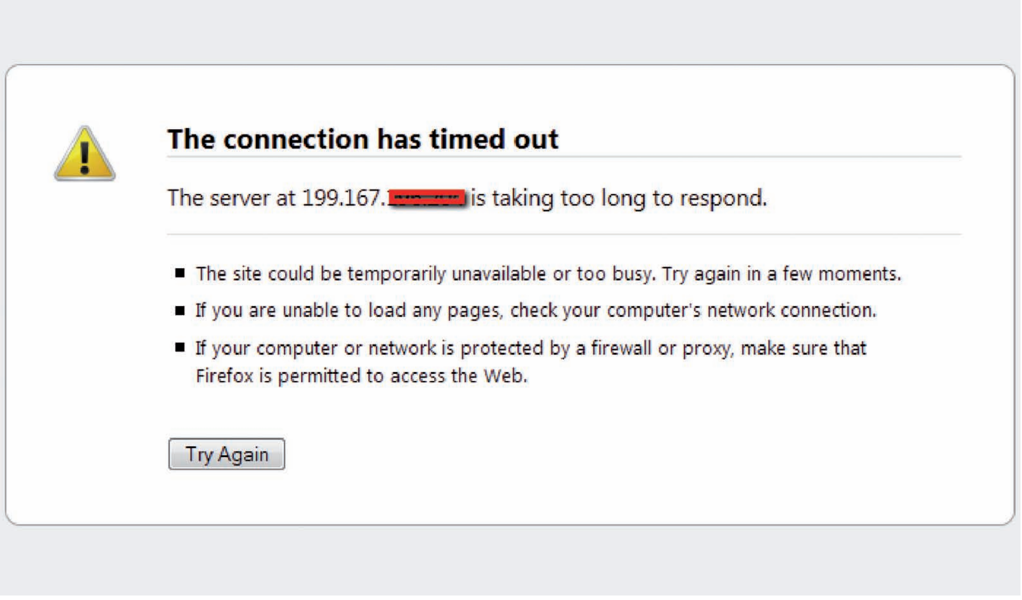
```
!udpflood 199.167.1XX.XX4 180
```

Αυτομάτως, τα πέντε VPS μας αρχίζουν την τρίλεπτη επίθεση τους. Από αυτά, ας σημειωθεί ότι δύο βρίσκονται στην Ελλάδα, ένα στην Ευρώπη κι άλλα δύο στην Αμερική. Ο στόχος μας βρίσκεται επίσης στην Αμερική.

Το πρώτο πράγμα που παρατηρούμε μετά από λίγη ώρα είναι το μήνυμα σφάλματος που ξαφνικά μας εμφανίζει το εργαλείο PuTTY, καθώς ήμασταν συνδεδεμένοι μέσω SSH στον στόχο. Καλό το πρώτο σημάδι. Περιμένουμε να φτάσουμε προς το τέλος της επίθεσης και προσπαθούμε να φορτώσουμε το περιβάλλον του εργαλείου ntop, από τον web server που τρέχει στον στόχο. Χμ, περιμένουμε και περιμένουμε... :D Καλό και το δεύτερο σημάδι και, ναι, μετά από λίγο παίρνουμε από τον Firefox το ωραιότατο μήνυμα ότι ο server-στόχος δεν ανταποκρίνεται.

Εδώ θα πρέπει να κάνουμε την εξής σημείωση. Δεν είναι βέβαιο ποιος πρόλαβε να βγάλει πρώτος το VPS εκτός λειτουργίας, εμείς ή ο host. Ως virtual machine, αυτός ο server ζει εντός άλλου, φυσικού μηχανήματος, κι επικοινωνεί με τον έξω κόσμο με εικονικές δικτυακές κάρτες, όπως

Να και το αποτέλεσμα της μικρής τελικής DDoS επίθεσής μας προς το VPS! Η συγκεκριμένη κατάσταση διήρκησε τουλάχιστον μισή ώρα. Ούτε ψύλλος στον κόρφο μας!



ακριβώς και τα virtual machines που στήνουμε με το VirtualBox ή το VMware. Παρατηρήσαμε ότι ενώ το VPS είχε στην ουσία “κλειδώσει” και δεν λειτουργούσε από “έξω”, μέσω του web panel της εταιρείας από την οποία το έχουμε αγοράσει μπορούσαμε να το διαχειριστούμε. Μετά από επικοινωνία μαζί τους φάνηκε πως έχουν ορισμένες δικλίδες ασφαλείας για τέτοιες περιπτώσεις, ώστε να μη δημιουργείται τεράστιος φόρτος στο συνολικό τους δίκτυο. Τελικά η λειτουργία του VPS αποκαταστάθηκε, όμως αυτό που μετράει τελικά είναι το αποτέλεσμα: Όλες οι διαδικτυακές υπηρεσίες που έτρεχαν σε αυτό ήταν απροσπέλαστες για τουλάχιστον 30 λεπτά *μετά* το πέρας της επίθεσης. Όχι και άσχημα :)

Επίλογος

Κάπου εδώ φτάνουμε στο τέλος της διαδρομής μας στον κόσμο των botnets -- τουλάχιστον στο συγκεκριμένο ταξίδι. Υπάρχουν βέβαια μπροστά μας αρκετές διασταυρώσεις και σκοτεινά μονοπάτια. Αυτή τη στιγμή τα botnets συγκαταλέγονται στις νούμερο ένα απειλές στο Διαδίκτυο και υπάρχει τεράστια κινητικότητα από εταιρείες και ειδικούς του χώρου, για την αντιμετώπιση του φαινομένου. Το ζήτημα αυτό θα σας απασχολήσει πολύ περισσότερο αργά ή γρήγορα, αρκεί να ασχολείστε με θέματα ασφαλείας.

Από τη μεριά μας είδαμε τη λειτουργία τους, θεωρητικά και πρακτικά. Στο πρακτικό μέρος βέβαια έχουμε κι άλλα πράγματα να κάνουμε. Για την ακρίβεια όχι εμείς, αλλά εσείς. Σας υποσχεθήκαμε μερικές ιδέες για την επέκταση του deltaBot. Υλοποιώντας τες θα ακονίσετε τις ικανότητες σας στον προγραμματισμό με την PHP, ενώ θα έχετε και την ικανοποίηση της βελτίωσης ενός προγράμματος ανοικτού κώδικα. Ας αναφέρουμε λοιπόν μερικές πιθανές επεκτάσεις ταξινομημένες ως προς τη δυσκολία τους.

Κατοχυρώστε το κανάλι της επιλογής σας και προστατέψτε το με κωδικό. Ακολούθως, τροποποιήστε την συνάρτηση join() ώστε να συμπεριλαμβάνει το key του καναλιού.

Επεκτάσεις και βελτιώσεις για την επίθεση UDP flood, όπως: Εμφάνιση στατιστικών μετά το πέρας της (π.χ., αριθμός πακέτων που στάλθηκαν και ταχύτητα αποστολής). Αντί για τη διάρκεια της επίθεσης να μπορεί να δοθεί και το πλήθος των πακέτων που θέλουμε να αποσταλούν. Τέλος, δυνατότητα επιλογής του μεγέθους των πακέτων.

Αποστολή mail από τα bots με χρήση της συνάρτησης mail() της PHP προς κάποια διεύθυνση, δίνοντας στο κανάλι το θέμα και το κείμενο του μηνύματος.

Στο port scanning, εκτός από θύρες χωρισμένες με κόμμα ο χρήστης να μπορεί να δίνει κι ένα διάστημα από θύρες (port range), με χρήση της παύλας (π.χ., !portscan IP 8-3000).

Εκτέλεση εντολών συστήματος στα απομακρυσμένα zombies και εμφάνιση των αποτελεσμάτων τους στο κεντρικό κανάλι συνομιλίας (π.χ., ls, cp, mkdir, touch κ.ά.)

Συζήτηση για όλα τα παραπάνω είναι καλοδεχούμενη και μπορείτε να χρησιμοποιήσετε τους γνωστούς τρόπους επικοινωνίας, από το site του περιοδικού: `http://deltahacker.gr`.

Εσείς τι δικαιολογία έχετε που η διαφήμισή σας απουσιάζει από εδώ;

Τολμήστε να μας την πείτε από το talk2us@deltahacker.gr. Όπως και να 'χει θα επικοινωνήσουμε κι εμείς μαζί σας - ή έστω με τον ανταγωνιστή σας ;)

Προς το παρόν είμαστε απασχολημένοι με τη δημιουργία του καλύτερου τεχνολογικού περιοδικού με θεματολογία ethical hacking. Ξέρετε. Για ανήσυχους, δημιουργικούς ανθρώπους που δεν βλέπουν τον υπολογιστή τους ως τοστιέρα. Σε πολλούς εξ αυτών αρέσει και το The Big Bang Theory. Και το Fringe, φυσικά.

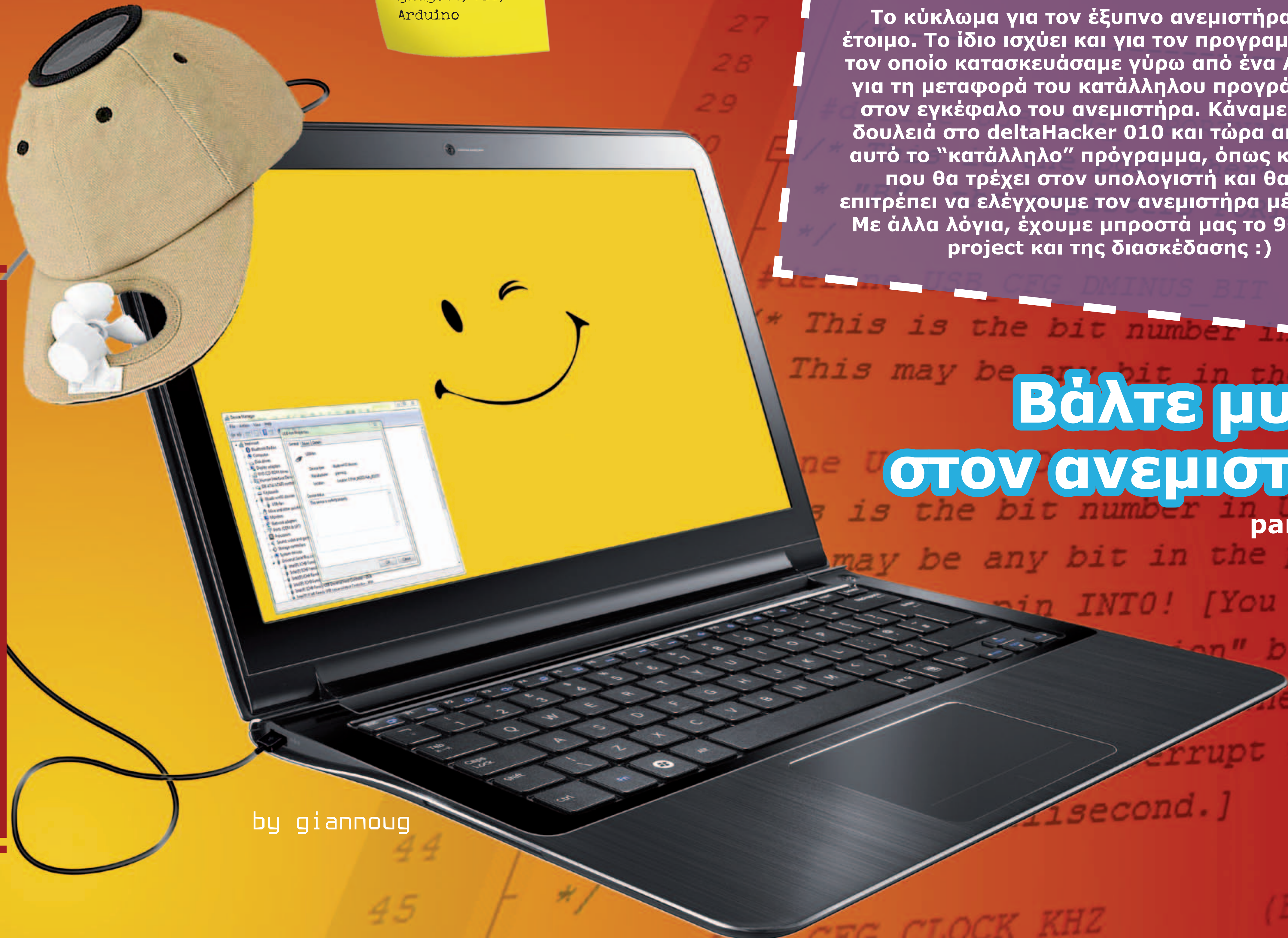
Skill: Intermediate

Tags: USB, gadgets, DIY, Arduino

Βάλτε μυαλό στον ανεμιστήρα

part 2

Το κύκλωμα για τον έξυπνο ανεμιστήρα είναι έτοιμο. Το ίδιο ισχύει και για τον προγραμματιστή, τον οποίο κατασκευάσαμε γύρω από ένα Arduino, για τη μεταφορά του κατάλληλου προγράμματος στον εγκέφαλο του ανεμιστήρα. Κάναμε πολλή δουλειά στο deltaHacker 010 και τώρα απομένει αυτό το "κατάλληλο" πρόγραμμα, όπως κι εκείνο που θα τρέχει στον υπολογιστή και θα μας επιτρέπει να ελέγχουμε τον ανεμιστήρα μέσω USB. Με άλλα λόγια, έχουμε μπροστά μας το 90% του project και της διασκέδασης :)



by giannoug

Στο προηγούμενο τεύχος του deltaHacker αποφασίσαμε να αναβαθμίσουμε έναν USB-powered ανεμιστήρα. Βλέπετε, είχαμε βαρεθεί να τον ανοιγοκλείνουμε από το διακόπτη του κι αποφασίσαμε να του χαρίσουμε λίγη ευφυΐα, ώστε να τον ελέγχουμε από τον υπολογιστή. Μέχρι στιγμής έχουμε εξετάσει το κύκλωμα που θα μας επιτρέψει να πετύχουμε κάτι τέτοιο και, φυσικά, έχουμε αναφέρει όλα τα απαραίτητα εξαρτήματα. Επομένως, το hardware γι’ αυτό το project είναι έτοιμο και μένει να εξετάσουμε το software. Συγκεκριμένα, σειρά έχει το πρόγραμμα για τον μικροελεγκτή (τον ATtiny) κι εκείνο για τον υπολογιστή. Το πρώτο θα επιτρέψει στον ATtiny να επικοινωνεί και να δέχεται εντολές μέσω USB, ενώ το δεύτερο θα φροντίσει για τον έλεγχο του ανεμιστήρα από το PC. Τέλος, θα πούμε δυο λόγια για την εύκολη και γρήγορη υλοποίηση του κυκλώματος πάνω σε πλακέτα και το γενικότερο σουλούπωμα του αναβαθμισμένου ανεμιστήρα μας. Πριν προχωρήσουμε, σημειώστε ότι μπορείτε να κατεβάσετε τον κώδικα των δύο προγραμμάτων από το

http://bit.ly/dh011usbfan

Στη συνέχεια θα εξετάσουμε τα πιο ενδιαφέροντα σημεία της κατασκευής των προγραμμάτων, αλλά αυτό δεν σημαίνει σε καμία περίπτωση ότι πρέπει να τα γράψετε από την αρχή.

Σοφτγουερική θύρα USB

Τεχνικά, το περισσότερο ενδιαφέρον συγκεντρώνεται στο πρόγραμμα που προορίζεται για τον μικροελεγκτή. Αυτό το πρόγραμμα στηρίζεται στη βιβλιοθήκη V-USB (Virtual USB), για να χαρίσει στον ATtiny μια θύρα USB! Μέσω αυτής της θύρας ο μικροελεγκτής θα επικοινωνεί με τον υπολογιστή, από τον οποίο θα δέχεται τις εντολές για τον έλεγχο του ανεμιστήρα. Η βιβλιοθήκη V-USB, λοιπόν, πετυχαίνει την πιο δύσκολη και ενδιαφέρουσα εργασία. Ας δούμε τη χρήση της από κοντά...

Προφανώς, το πρώτο βήμα είναι η λήψη του κώδικα της βιβλιοθήκης από τον σχετικό δικτυακό τόπο:

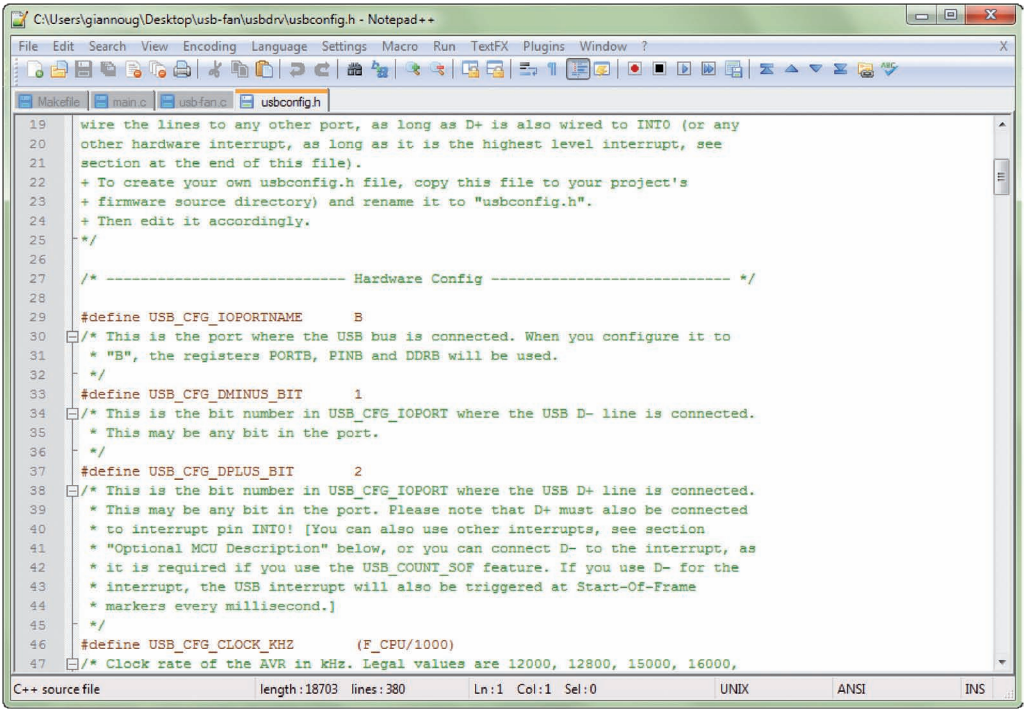
www.obdev.at/products/vusb/download.html

Αφού αποσυμπιέσουμε το πακέτο, πρέπει να αντιγράψουμε ολόκληρο τον κατάλογο usbdrv μέσα στον φάκελο όπου θα αναπτύξουμε το πρόγραμμά μας. Ο κατάλογος usbdrv περιέχει πολλά αρχεία, με άφθονο κώδικα και πολλές ρυθμίσεις. Εμείς πρέπει να σταθούμε στο usbconfig-prototype.h, το οποίο περιέχει όλες τις παραμέτρους για τη βασική ρύθμιση της βιβλιοθήκης. Το μετονομάζουμε (ή αντιγράφουμε) σε usbconfig.h και η περιπέτεια με τη βιβλιοθήκη V-USB ξεκινά! Για αρχή, πρέπει να καθορίσουμε ποιους ακροδέκτες του μικροελεγκτή θα χρησιμοποιεί η βιβλιοθήκη για την εικονική θύρα USB:

```
#define USB_CFG_IOPORTNAME          B
#define USB_CFG_DMINUS_BIT          1
#define USB_CFG_DPLUS_BIT           2
```

Όπως βλέπετε στις παραπάνω γραμμές, επιλέξαμε τα pin 1 και 2 της θύρας PORTB[1]. Συγκεκριμένα, για το σήμα DMINUS επιλέξαμε το pin 1 και για το σήμα DPLUS το pin 2. Αυτή η λεπτομέρεια μοιάζει ασήμαντη αλλά είναι ιδιαίτερα κρίσιμη. Βλέπετε, όταν ξεμπερδέψουμε με το πρόγραμμα ο μικροελεγκτής θα έχει αποκτήσει όλο το μηχανισμό μιας θύρας USB -- σε software. Ωστόσο θα του λείπει η φυσική θύρα USB. Γι’ αυτό το λόγο, θα

1. Πρόκειται για τη μοναδική θύρα του ATtiny45.



Όλες οι απαραίτητες ρυθμίσεις για τη λειτουργία της βιβλιοθήκης V-USB βρίσκονται στο αρχείο usbconfig.h.

χρειαστεί να συνδέσουμε πάνω του ένα καλώδιο ή μια υποδοχή USB. Τα καλώδια USB περιλαμβάνουν τέσσερα λεπτότερα καλώδια, με τυποποιημένο χρωματισμό (κόκκινο, μαύρο, πράσινο και λευκό). Το πώς συνδέονται αυτά στο κύκλωμά μας φαίνεται στις σχετικές εικόνες, στο πρώτο μέρος του άρθρου (σελίδα 97, τεύχος 010). Τώρα, αν κάποιος είναι αρκετά μερακλής και σκοπεύει να συνδέσει στο κύκλωμα μια υποδοχή USB, θα χρειαστεί λίγη παραπάνω προσοχή. Η επαφή DMINUS της υποδοχής πρέπει να συνδεθεί στο άσπρο καλώδιο και η επαφή DPLUS στο πράσινο. Φυσικά, πρέπει να συνδεθούν και οι άλλες δύο επαφές της υποδοχής (+5V και GND), εφόσον το κύκλωμά μας θα τροφοδοτείται από το δίαυλο USB.

Μετά από αυτή την αναγκαία (αλλά σύντομη) παρένθεση, μπορούμε να επιστρέψουμε στη ρύθμιση της βιβλιοθήκης. Στο αρχείο usbconfig.h θα βρούμε και μια γραμμή σαν την ακόλουθη:

```
#define USB_CFG_MAX_BUS_POWER 500
```

Όπως έχουμε πει, η κατασκευή μας θα τροφοδοτείται αποκλειστικά από τη θύρα USB. Ωστόσο, οι θύρες USB δίνουν το μέγιστο δυνατό ρεύμα[2] μέχρι να γίνει το negotiation με την εκάστοτε συσκευή. Αμέσως μετά η παροχή ρεύματος περιορίζεται στο επίπεδο που καθορίζει η συνδεδεμένη συσκευή. Σύμφωνα με μετρήσεις που κάναμε, ο ανεμιστήρας μας απαιτεί περί τα 250mA. Ωστόσο, για λόγους συμβατότητας με περισσότερους ανεμιστήρες USB και για να μην παίζουμε με τα όρια της εκάστοτε θύρας, αποφασίσαμε να ορίσουμε το USB_CFG_MAX_BUS_POWER στα 500mA.

Στη συνέχεια πρέπει να βρούμε τις ακόλουθες γραμμές και τις ενεργο-

2. Συνήθως 500mA, αν και υπάρχουν μητρικές με "ενισχυμένες" θύρες USB (τις ονομάζουν και θύρες φόρτισης), οι οποίες μπορούν να δώσουν έως και 1A!

ποιήσουμε, αφαιρώντας τους χαρακτήρες που οριοθετούν ένα comment block ("/*" και "*/"):

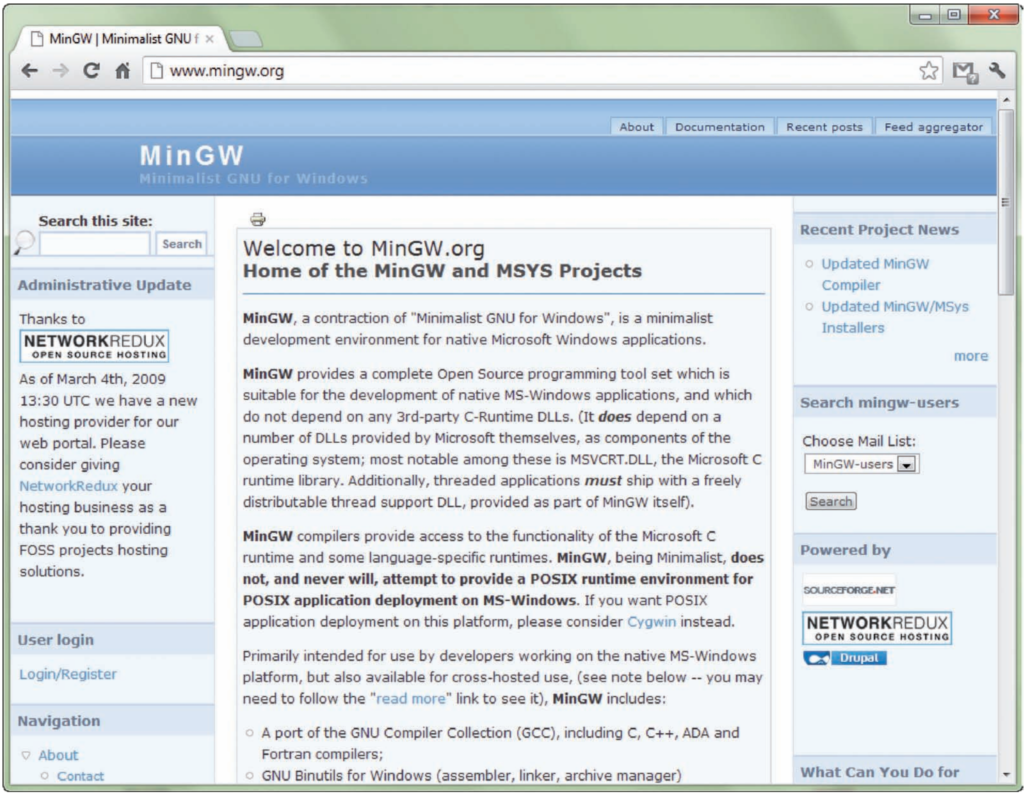
```
#define USB_RESET_HOOK(resetStarts)  if(!resetStarts){hadUsbReset();}
#ifndef __ASSEMBLER__
extern void hadUsbReset(void);
#endif
```

Ουσιαστικά, με αυτή την κίνηση ενεργοποιούμε την αυτόματη εκτέλεση της συνάρτησης hadUsbReset, κατά την πραγματοποίηση μιας επανεκκίνησης της εικονικής συσκευής USB. Αυτή η συνάρτηση, όπως θα δούμε αργότερα, πραγματοποιεί μια λεπτομερή ρύθμιση (finetune) του ρολογιού του μικροελεγκτή. Αυτή η ρύθμιση είναι αναγκαία, καθώς ο κώδικας της βιβλιοθήκης πρέπει να παράγει τα διάφορα πακέτα δεδομένων με τον ακριβή ρυθμό που επιβάλουν οι προδιαγραφές του προτύπου USB. Η συνάρτηση hadUsbReset θα κάνει χρήση της συνάρτησης usbMeasureFrameLength. Έτσι, πρέπει να βρούμε τη σχετική ακόλουθη γραμμή στο αρχείο ρυθμίσεων και να επιβεβαιώσουμε ότι τελειώνει μ' έναν άσσο:

```
#define USB_CFG_HAVE_MEASURE_FRAME_LENGTH      1
```

Κάθε συσκευή USB ταυτοποιείται από δύο αναγνωριστικά μεγέθη. Αυτά τα μεγέθη (vendor_id και device_id) δεν δηλώνονται τυχαία από τον εκάστοτε κατασκευαστή, αλλά κατόπιν συνεννόησης με την οργάνωση που διαχειρίζεται το πρωτόκολλο και, φυσικά, αφού πληρωθεί η σχετική συν-

Το MinGW προσφέρει τη σουίτα μεταγλωττιστών GCC για το περιβάλλον των Windows. Με τον GCC θα μεταγλωττίσουμε τα δύο προγράμματά μας: αυτό για τον ATtiny κι εκείνο για τον υπολογιστή.



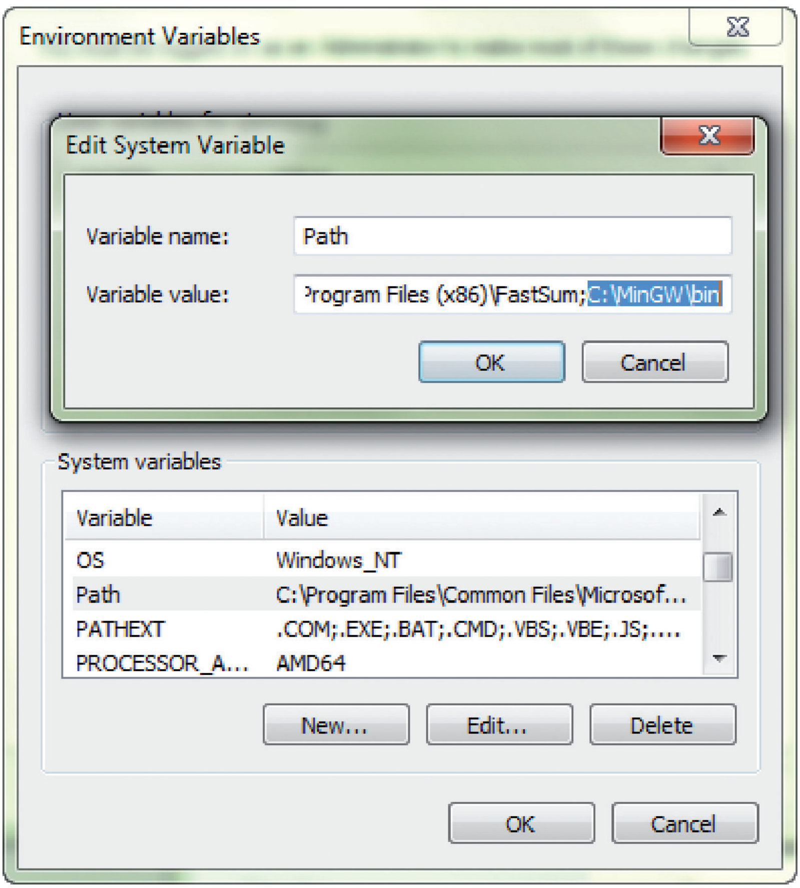
δρομή! Ευτυχώς για 'μάς, οι άνθρωποι πίσω από το V-USB έχουν κατοχυρώσει ορισμένα ID, τα οποία διαθέτουν στους χρήστες της βιβλιοθήκης δωρεάν και σχεδόν ελεύθερα[3]. Επομένως, όταν συναντήσουμε τις παρακάτω γραμμές, θα πρέπει να τις αφήσουμε ως έχουν:

```
#define  USB_CFG_VENDOR_ID          0xc0, 0x16
#define  USB_CFG_DEVICE_ID          0xdc, 0x05
#define  USB_CFG_DEVICE_VERSION     0x00, 0x01
```

Ωστόσο, λίγο χαμηλότερα δηλώνονται δύο πίνακες στους οποίους επιβάλλεται να επέμβουμε. Ο ένας πίνακας καθορίζει το όνομα του κατασκευαστή και ο άλλος το όνομα της συσκευής. Εμείς τροποποιήσαμε ως εξής:

```
#define USB_CFG_VENDOR_NAME          'g', 'i', 'a', 'n', 'n', 'o', 'u', 'g'
#define USB_CFG_VENDOR_NAME_LEN      8
#define USB_CFG_DEVICE_NAME          'U', 'S', 'B', '-', 'f', 'a', 'n'
#define USB_CFG_DEVICE_NAME_LEN      7
```

Το πρόγραμμα που θα εξετάσουμε αργότερα, αυτό για τον υπολογιστή, θα αναγνωρίζει τη συσκευή μας από αυτά τα δύο μεγέθη. Επομένως, θα



Μετά την εγκατάσταση του MinGW, θα ήταν καλό να προσθέσουμε στο PATH του συστήματος, τον κατάλογο εγκατάστασής του...

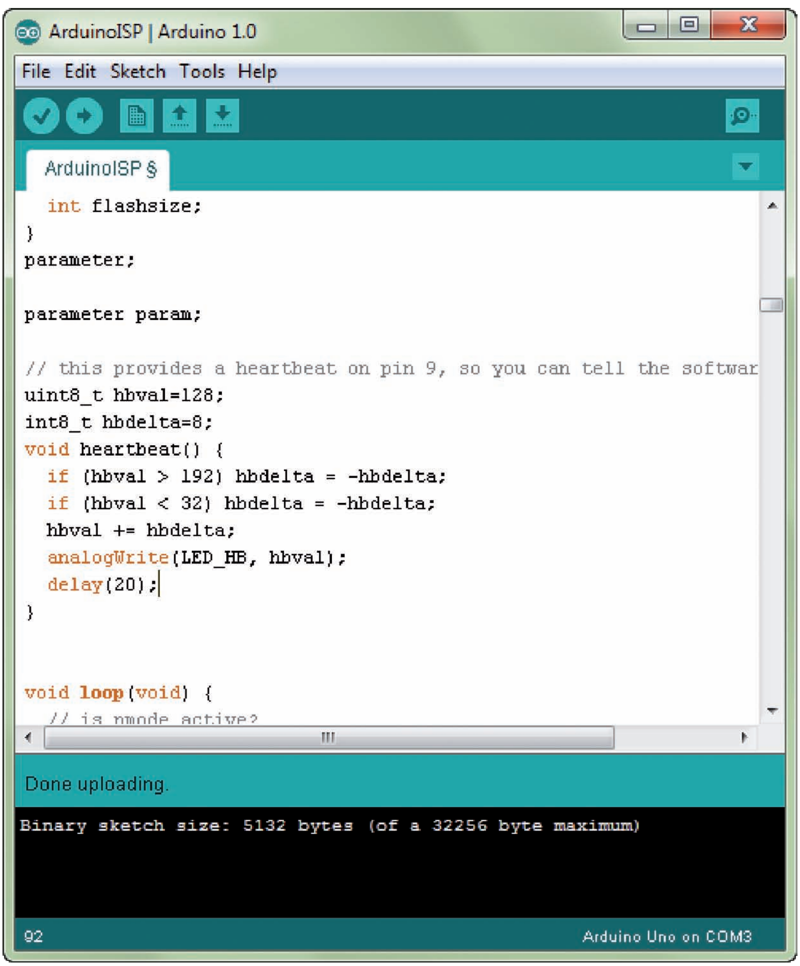
3. Οι όροι χρήσης αναφέρονται στο αρχείο USB-IDs-for-free.txt.

πρέπει να είμαστε προσεκτικοί κατά τη δήλωσή τους και φυσικά να θυμόμαστε τις τιμές τους.

Κώδικας για τον ATtiny

Μέχρι στιγμής έχουμε πει τα πάντα για τη ρύθμιση της βιβλιοθήκης και μπορούμε να εξετάσουμε το πρόγραμμα που την αξιοποιεί. Σε αυτό το σημείο, θα ήταν χρήσιμο να ανοίξετε το πακέτο με τον κώδικα που σας δώσαμε. Το πρόγραμμα για τον μικροελεγκτή ονομάζεται main.c και, όπως θα διαπιστώσετε, ο κώδικας χωρίζεται σε τέσσερα ευδιάκριτα τμήματα. Τα δύο από αυτά θα σας φανούν γνώριμα, ενώ τα άλλα δύο σχετίζονται με τη βιβλιοθήκη V-USB και θα τα δείτε για πρώτη φορά. Ας ξεκινήσουμε με τα γνώριμα...

Οι πρώτες γραμμές ενός προγράμματος, σε όλες τις γλώσσες, αξιοποιούνται λίγο-πολύ με τον ίδιο τρόπο. Σε αυτές ενσωματώνονται οι απαιτούμενες βιβλιοθήκες και δηλώνονται διάφορες σταθερές (constants). Στο πρόγραμμά μας, οι σταθερές αποτελούν τα δύο πιθανά μηνύματα που θα



Αν κατά το ανέβασμα του προγράμματος συναντήσουμε προβλήματα, θα χρειαστεί να ανατρέξουμε στο άρθρο του deltaHacker 010 και να κάνουμε μια αλλαγή στο Sketch του Arduino ISP. Στη συνάρτηση heartbeat αλλάζουμε το delay(40) σε delay(20)...

```
C:\Windows\system32\cmd.exe
avrdude: writing flash (1914 bytes):

Writing | ################################################## | 100% 2.42s

avrdude: 1914 bytes of flash written
avrdude: verifying flash memory against main.hex:
avrdude: load data flash data from input file main.hex:
avrdude: input file main.hex auto detected as Intel Hex
avrdude: input file main.hex contains 1914 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 2.46s

avrdude: verifying ...
avrdude: 1914 bytes of flash verified

avrdude: safemode: lfuse reads as E1
avrdude: safemode: hfuse reads as DD
avrdude: safemode: efuse reads as FF
avrdude: safemode: Fuses OK

avrdude done.  Thank you.

C:\Users\giannoug\Desktop\avr_usb>
```

Οι κατάλληλες τιμές για τα fuses του ATtiny45, γράφτηκαν με επιτυχία!

μπορεί να στέλνει ο υπολογιστής στον ATtiny, για τον έλεγχο του ανεμιστήρα. Όσο για τις βιβλιοθήκες, η χρησιμότητα όσων ενσωματώνουμε είναι μάλλον προφανής. Το δεύτερο γνώριμο τμήμα αποτελεί η συνάρτηση main, από την οποία ξεκινά πάντα η εκτέλεση ενός προγράμματος σε C. Στην περίπτωσή μας, η εν λόγω συνάρτηση αρχικοποιεί τη βιβλιοθήκη V-USB, καθώς και το watchdog timer του μικροελεγκτή. Ώπα! Καλά η βιβλιοθήκη, θα πει κανείς, αλλά το watchdog timer τι είναι και τι ρόλο βαράει; Πρόκειται για έναν μετρητή (timer) υλοποιημένο στο hardware του μικροελεγκτή, ο οποίος χρησιμοποιείται για την αντιμετώπιση των κολλημάτων. Ο συγκεκριμένος μετρητής μετράει διαρκώς από το μηδέν ως τη μέγιστη τιμή του και όταν φτάνει σε αυτήν, πραγματοποιεί ένα system reset. Ο μόνος τρόπος για να *μην* φτάσει στη μέγιστη τιμή και να *μην* κάνει reset στον μικροελεγκτή, είναι να του μιλήσει το πρόγραμμα. Έτσι, ένα σωστά γραμμένο πρόγραμμα φροντίζει να μιλάει στο watchdog timer συχνά και σε τακτά χρονικά διαστήματα. Αν το πρόγραμμα κολλήσει ο watchdog timer υπερχειλίζει (φτάνει στη μέγιστη τιμή του και την ξεπερνά) κι ακολούθως κάνει reset στο σύστημα. Έτσι, ο μικροελεγκτής δεν μένει ποτέ κολλημένος!

Μετά τις δύο αρχικοποιήσεις που αναφέραμε παραπάνω, η συνάρτηση main πραγματοποιεί μια αποσύνδεση και επανασύνδεση της εικονικής συσκευής USB (USB re-enumeration). Αυτό γίνεται να για να βεβαιωθούμε ότι η σύνδεση δεν είναι "προβληματική" και δεν ξεκίνησε πριν από κάποιο reset ή κόλλημα του συστήματος. Η συνάρτηση ολοκληρώνεται με έναν ατέρμονα βρόχο, ο οποίος ελέγχει τη σύνδεση USB για εισερχόμενα πακέτα και φροντίζει να μιλάει στο watchdog timer -- για να του λέει ότι είναι όλα καλά.

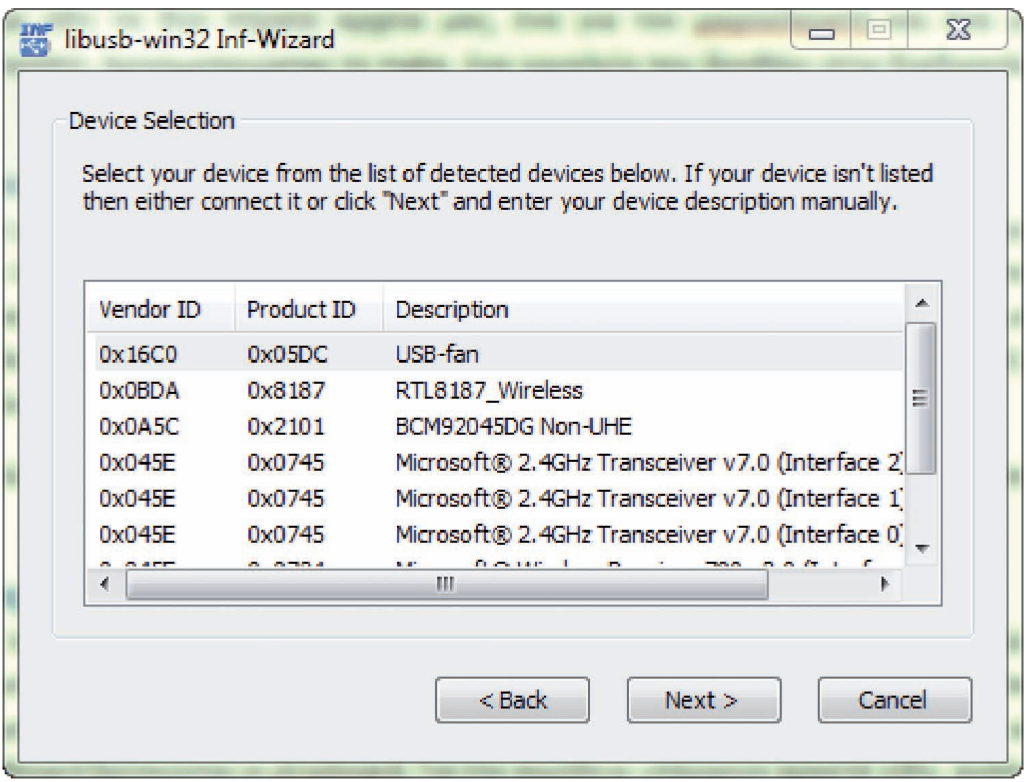
V-USB σε δράση

Τα δύο τμήματα του προγράμματος, με τα οποία δεν έχουμε ασχοληθεί ακόμα, πετυχαίνουν τις πιο ενδιαφέρουσες λειτουργίες. Η συνάρτηση usbFunctionSetup επεξεργάζεται τα δεδομένα που λαμβάνει ο μικροελεγκτής από την εικονική θύρα USB. Ουσιαστικά, από όλο το πακέτο των byte που περιλαμβάνει ένα αίτημα του πρωτοκόλλου USB, η συνάρτηση

εστιάζει στην εντολή (στο custom request) που στέλνει το πρόγραμμα του υπολογιστή και πράττει τα δέοντα. Με απλά λόγια, ανάβει ή σβήνει τον ανεμιστήρα, σύμφωνα με τις οδηγίες που λαμβάνει από τον υπολογιστή. Ο έλεγχος του ανεμιστήρα επιτυγχάνεται από το FAN_PIN[4] της θύρας PORTB. Αν ρίξετε μια ματιά στο κύκλωμα, θα δείτε ότι ακριβώς εκεί συνδέεται το τρανζίστορ μας (TIP120), σε ρόλο διακόπτη.

Για το τέλος αφήσαμε τη συνάρτηση hadUsbReset. Εδώ πραγματοποιείται μια μικρορύθμιση στο ρολόι του μικροελεγκτή, ώστε η συχνότητα λειτουργίας του και κατ' επέκταση η ταχύτητα εκτέλεσης της βιβλιοθήκης να βρίσκεται εντός των προδιαγραφών που προβλέπει το πρότυπο USB. Η επέμβαση στη συχνότητα λειτουργίας του συστήματος επιτυγχάνεται με έναν ειδικό μηχανισμό που ενσωματώνουν οι μικροελεγκτές AVR, οι οποίοι διαθέτουν ενσωματωμένο ρολόι. Συγκεκριμένα, η μικρορύθμιση της συχνότητας πραγματοποιείται με την απόδοση της κατάλληλης τιμής στον καταχωρητή OSCCAL (Oscillator Calibration). Η συνάρτηση hadUsbReset, λοιπόν, πραγματοποιεί μια σειρά μετρήσεων της ταχύτητας, με τη βοήθεια της συνάρτησης usbMeasureFrameLength (την είχαμε ενεργοποιήσει κι αυτήν κατά τη ρύθμιση της βιβλιοθήκης). Πριν από κάθε μέτρηση, το πρόγραμμα μεταβάλλει την τιμή του OSCCAL και σιγά-σιγά, επιτυγχάνει τη βέλτιστη ρύθμισή του. Οι λεπτομέρειες γύρω από αυτή τη διαδικασία ξεφεύγουν κατά πολύ από το θέμα μας και δεν είναι της παρούσης. Σημειώστε μόνο ότι η συγκεκριμένη συνάρτηση χρησιμοποιείται σε κάθε project της V-USB, όταν ο μικροελεγκτής χρονίζεται από το

Είμαστε έτοιμοι να φτιάξουμε τους κατάλληλους drivers για την συσκευή μας. Για το σκοπό αυτό εκτελούμε το inf-wizard.exe από τον κατάλογο της βιβλιοθήκης libusb και αρχικά προσδιορίζουμε τη συσκευή μας...



4. Μια ακόμα σταθερά που ορίστηκε στην αρχή του προγράμματος. Το FAN_PIN αντιστοιχεί στο PB3...

ενσωματωμένο ρολόι και όχι από κάποιον κρύσταλλο.

Στον υπολογιστή

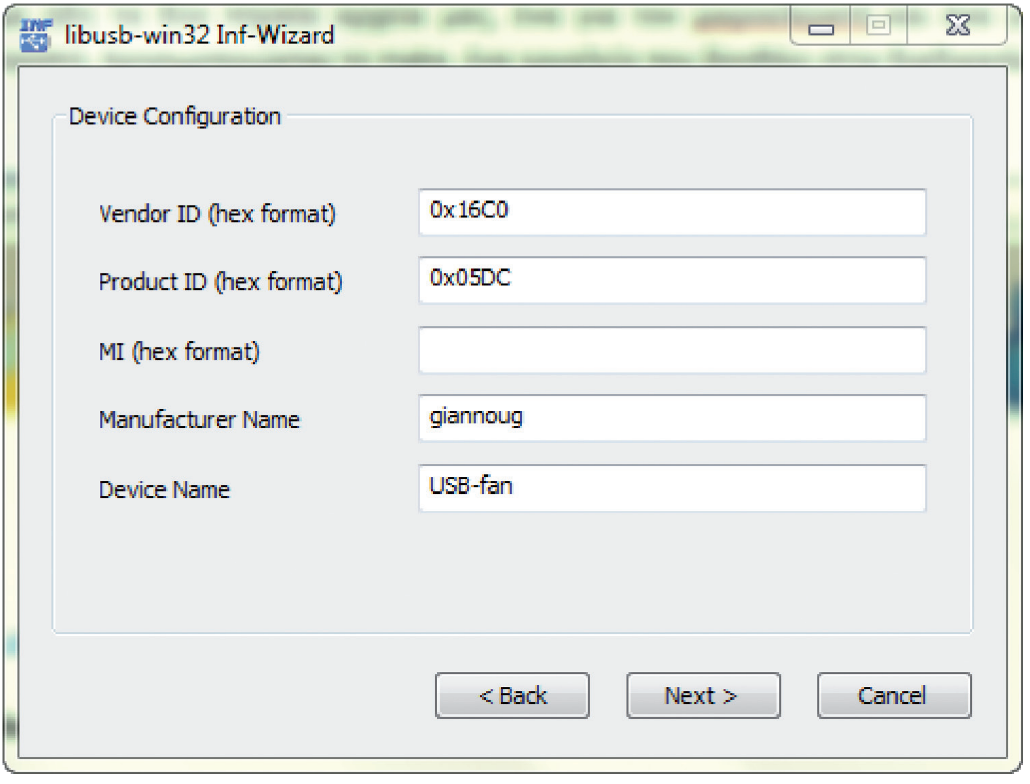
Έχοντας έτοιμο το πρόγραμμα για τον ATtiny, μένει να δούμε το πρόγραμμα που θα τρέχει στον υπολογιστή. Προφανώς, οι θύρες USB είναι έτοιμες στον υπολογιστή και δεν θα τις υλοποιήσουμε σε software. Ωστόσο, θα χρειαστούμε και πάλι μια βιβλιοθήκη για τον εύκολο χειρισμό του πρωτοκόλλου USB, μέσα από το πρόγραμμά μας. Συγκεκριμένα, θα χρειαστούμε τη βιβλιοθήκη libusb. Μπορείτε να βρείτε την πιο πρόσφατη έκδοσή της στο

http://sourceforge.net/projects/libusb-win32/files/libusb-win32-releases

Στη δική μας περίπτωση, η πλέον ενημερωμένη έκδοση ήταν η 1.2.6.0 και γι' αυτό κατεβάσαμε το πακέτο libusb-win32-bin-1.2.6.0.zip. Πριν προχωρήσουμε, σημειώστε ότι ο κώδικας για το πρόγραμμα που θα τρέχει στον υπολογιστή βρίσκεται στο αρχείο usb-fan.c. Κατά τα γνωστά, θα ήταν καλό να το ανοίξετε και να του ρίξετε καμιά ματιά, αν όχι να το μελετήσετε διεξοδικά.

Το πρόγραμμά μας ακολουθεί τη λογική που συναντά κανείς στις περισσότερες εφαρμογές της βιβλιοθήκης libusb. Αρχικά, προσπαθεί να εντοπίσει τη συσκευή μας, χρησιμοποιώντας τα vendor name και device name που δηλώσαμε κατά τη ρύθμιση της V-USB ("giannoug" και "usb-fan"). Αυτός ο εντοπισμός πραγματοποιείται εντός της συνάρτησης usbOpenDevice.

Προσέχουμε ώστε να αλλάξουμε το Manufacturers name σε αυτό που ορίσαμε στη βιβλιοθήκη V-USB!

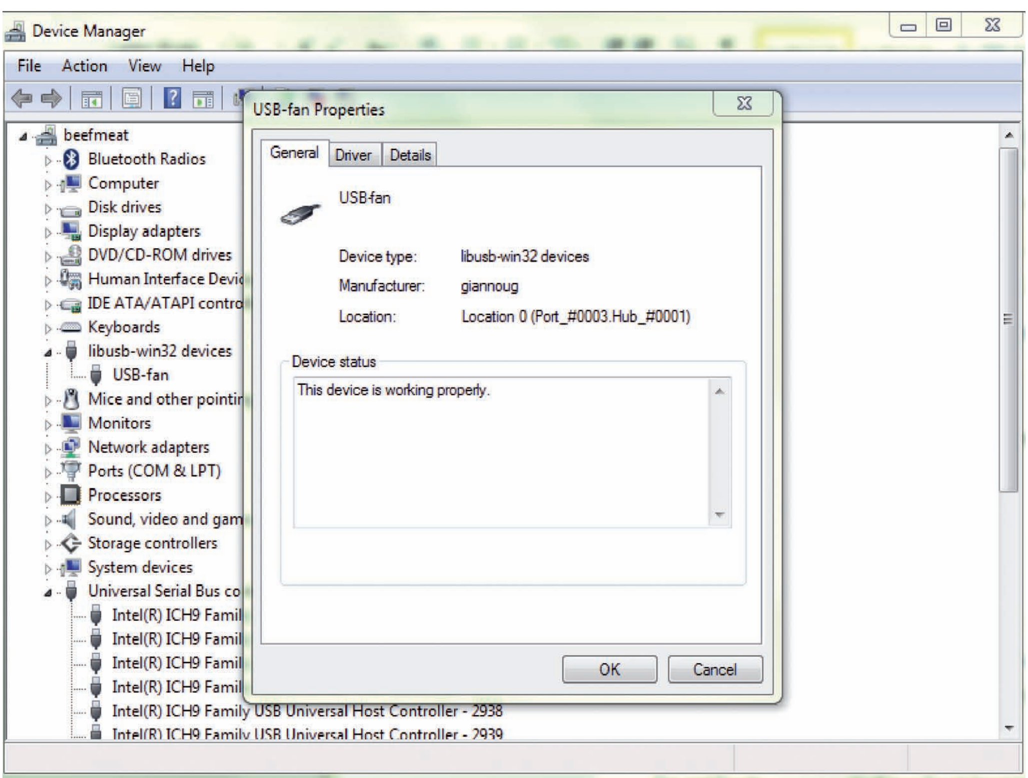


Εκεί, αφού δημιουργηθεί μια λίστα με όλους τους διαύλους USB του συστήματος και όλες τις συσκευές, ξεκινά ένας βρόχος που σαρώνει ένα προς ένα τα στοιχεία της λίστας. Για κάθε στοιχείο της καλείται η συνάρτηση usbGetDescriptorString, η οποία λαμβάνει το vendor name και το device name. Αμέσως μετά, τα δύο μεγέθη συγκρίνονται με τις τιμές που έχουμε δηλώσει (ξανά: "giannoug" και "fan-usb"). Αν οι τιμές ταυτίζονται, η εκτέλεση του βρόχου κι ολόκληρης της συνάρτησης usbOpenDevice τερματίζεται, καθώς η συσκευή μας έχει εντοπιστεί. Σε διαφορετική περίπτωση ο βρόχος συνεχίζει να εκτελείται, μέχρι να ελεγχθούν όλες οι συσκευές USB. Με αυτό τον απλοϊκό αλλά εγγυημένο τρόπο λειτουργίας, το πρόγραμμά εντοπίζει τον έξυπνο ανεμιστήρα μας και συνδέεται. Τέλος, με τη συνάρτηση usb_control_msg (ανήκει στη βιβλιοθήκη libusb) το πρόγραμμα στέλνει στον ανεμιστήρα το κατάλληλο σήμα (USB_FAN_ON ή USB_FAN_OFF), σύμφωνα με την παράμετρο που δώσαμε κατά την εκτέλεσή του.

Εργαλεία και μεταγλώττιση

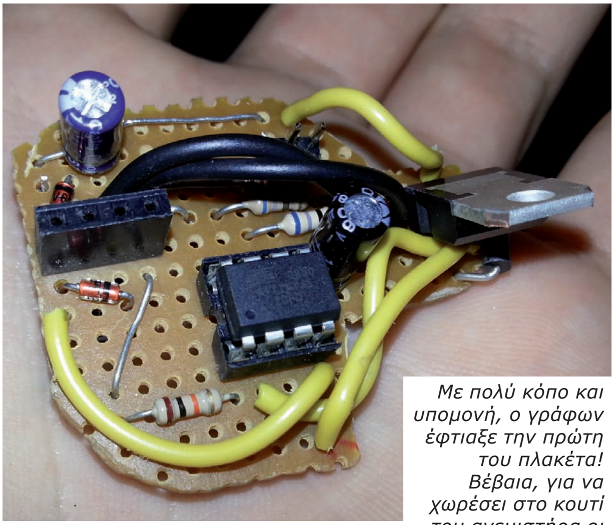
Η μεταγλώττιση των δύο προγραμμάτων μας γίνεται με τον GCC. Βέβαια, για τη μεταγλώττιση του προγράμματος που θα τρέχει στον ATtiny χρειαζόμαστε μια ειδική εκδοχή του GCC. Για το σκοπό αυτό, όπως αναφέραμε και στο πρώτο μέρος του άρθρου, θα πρέπει να κατεβάσουμε και να εγκαταστήσουμε το WinAVR:

`http://sourceforge.net/projects/winavr/files/WinAVR`



Όλα πήγαν καλά! Η συσκευή μας λειτουργεί, οι drivers το ίδιο και το πρόγραμμα είναι ικανό να τη χειριστεί όπως περιμέναμε :)

Με την εγκατάσταση του εν λόγω προγράμματος θα προστεθούν διάφορα εργαλεία στο σύστημά μας, μεταξύ των οποίων το make και το avrdude. Το πρώτο θα μας βοηθήσει να αυτοματοποιήσουμε τη μεταγλώττιση των δύο προγραμμάτων, ενώ το δεύτερο θα μας βοηθήσει να μεταφέρουμε το μεταγλωττισμένο πρόγραμμα στον μικροελεγκτή. Το καλό με το WinAVR (αν και σε κάποιους δεν αρέσουν οι πρωτοβουλίες) είναι ότι τροποποιεί αυτόματα το PATH του συστήματος. Έτσι, μετά την εγκατάστασή του, μπορούμε να ανοίξουμε το command line των Windows (cmd.exe) και να τρέξουμε το make, το avrdude, το avr-gcc κ.λπ., χωρίς να δηλώσουμε τη θέση τους. Για τη μεταγλώττιση του προγράμματος που θα τρέχει στον υπολογιστή χρειαζόμαστε τον "κανονικό" GCC. Ένας εύκολος τρόπος για να τον αποκτήσουμε στα Windows, είναι να εγκαταστήσουμε το MinGW, με τις προεπιλεγμένες ρυθμίσεις. Η πιο πρόσφατη έκδοση του εν λόγω πακέτου βρίσκεται πάντα στη διεύθυνση



Με πολύ κόπο και υπομονή, ο γράφων έφτιαξε την πρώτη του πλακέτα! Βέβαια, για να χωρέσει στο κουτί του ανεμιστήρα οι άκρες της πλακέτας χρειάστηκαν λίγη ...περιποίηση!

`http://sourceforge.net/projects/mingw`

Εδώ χρειάζεται λίγη προσοχή. Το MinGW δεν τροποποιεί αυτόματα τη μεταβλητή περιβάλλοντος (environment variable) PATH. Επομένως, για να μπορούμε να καλούμε τον GCC από τη γραμμή εντολών και χωρίς να δηλώνουμε κάθε φορά τη θέση του, πρέπει να κάνουμε την κατάλληλη προσθήκη στη μεταβλητή PATH χειροκίνητα. Για το σκοπό αυτό αρκεί ένα δεξί κλικ στον "Υπολογιστή" και επιλογή του "Ιδιότητες". Κατόπιν, στο παράθυρο που εμφανίζεται πρέπει να κάνουμε κλικ στο "Ρυθμίσεις συστήματος για προχωρημένους". Από εκεί, αρκεί ένα κλικ στο κουμπί "Μεταβλητές περιβάλλοντος". Τέλος, πρέπει να προσθέσουμε στη τιμή της PATH κάτι τέτοιο ";<MinGW installation-directory>\bin". Για τις προεπιλεγμένες ρυθμίσεις εγκατάστασης του MinGW, η θέση εγκατάστασης είναι ο φάκελος C:\MinGW. Επομένως, στη μεταβλητή PATH θα πρέπει να προσθέσουμε το ";C:\MinGW\bin".



Η κάτω πλευρά της πλακέτας. Φαίνεται μια τομή στις γραμμές του stripboard. Αυτή η διακοπή στις γραμμές ήταν αναγκαία ώστε να μη βραχυκυκλώνουν οι ακροδέκτες του ATtiny. Όσο για τους πυκνωτές και την παράξενη τοποθέτησή τους, ξέρετε τι λένε: Ουδέν μονιμότερο του προσωρινού!

Μετά από τα παραπάνω, μπορούμε να μεταβούμε εκεί όπου έχουμε αποθηκεύσει τα δύο προγράμματα. Όπως θα δείτε στο σχετικό πακέτο που σας δώσαμε, στην ίδια θέση υπάρχει και ένα MakeFile. Επομένως, για την αυτόματη και εύκολη μεταγλώττιση των προγραμμάτων αρκεί να ανοίξουμε ένα παράθυρο με τη γραμμή εντολών, να με-

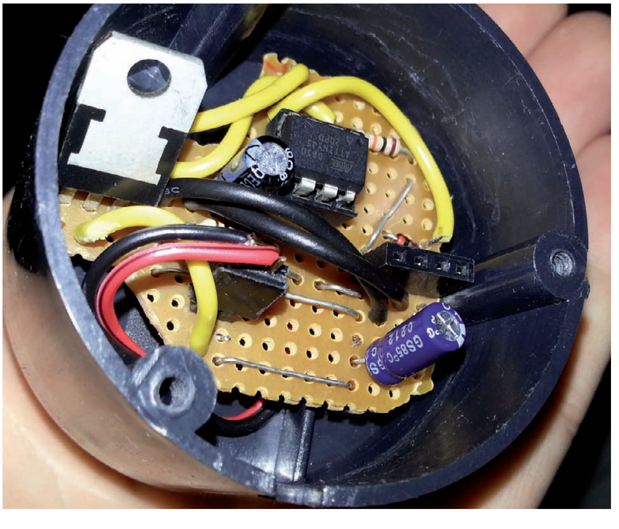
ταβούμε στο φάκελο με τα προγράμματα και να τρέξουμε το make. Το MakeFile που έχουμε κατασκευάσει θα φροντίσει για τη μεταγλώττιση των δύο προγραμμάτων, ενώ μετά θα φτιάξει και δύο απλά scripts για τον εύκολο χειρισμό του usb-fan.exe. Συγκεκριμένα, θα φτιάξει δύο batch files, για την εκκίνηση ή την παύση του ανεμιστήρα μας από τον υπολογιστή!

Δυο βήματα ακόμα

Μπορεί να μην έχουμε ξεμπερδέψει ακόμα, αλλά μην ανησυχείτε! Απομένουν μόνο δύο απλές εργασίες. Κατ' αρχάς, πρέπει να ρυθμίσουμε κατάλληλα το hardware του ATtiny45 και στη συνέχεια να γράψουμε στον αποθηκευτικό του χώρο το πρόγραμμά μας. Στο πρώτο μέρος του άρθρου, αναφέραμε ότι αυτές οι εργασίες προϋποθέτουν την αγορά ενός προγραμματιστή για AVR ή την κατασκευή ενός -- με τη βοήθεια του Arduino. Προφανώς, η δεύτερη λύση είναι ευκολότερη και φτηνότερη. Ακριβώς γι' αυτό, την είχαμε παρουσιάσει αναλυτικά και είχαμε περιγράψει όλες τις συνδέσεις που πρέπει να γίνουν στο Arduino. Τώρα, λοιπόν, έφτασε η στιγμή να χρησιμοποιήσουμε αυτόν τον προγραμματιστή. Για το σκοπό αυτό, αρκεί να εκτελέσουμε μια φορά το make με παράμετρο το fuses και μία ακόμα με παράμετρο το flash. Με την πρώτη εκτέλεση του make, το avrdude θα ενεργοποιήσει το ενσωματωμένο σύστημα χρονισμού του ATtiny45, καθώς και τον watchdog timer (αυτό επιτυγχάνεται με την κατάλληλη ρύθμιση των λεγόμενων fuses του ATtiny). Με τη δεύτερη εκτέλεση του make θα εκτελεστεί πάλι ο avrdude, για να γράψει το πρόγραμμά μας στον μικροελεγκτή. Όπως αντιλαμβάνεστε, η εκτέλεση του avrdude με τις κατάλληλες κάθε φορά παραμέτρους, περιγράφεται εντός του MakeFile. Αυτές τις παραμέτρους δεν θα χρειαστεί να τις πειράξουμε, παρά μόνο σε μία περίπτωση. Αν *δεν* εμφανίζεται η κλασική σειραϊκή θύρα COM3, όταν συνδέουμε το Arduino στον υπολογιστή, θα πρέπει να ανοίξουμε το MakeFile και να τροποποιήσουμε την τιμή της μεταβλητής PROGRAMMER. Για παράδειγμα, αν η σύνδεση του Arduino εμφανίζει στον υπολογιστή μας τη θύρα COM4, η εν λόγω μεταβλητή θα πρέπει να γίνει κάπως έτσι:

```
PROGRAMMER = -P com4 -b 19200 -c avrisp
```

Με αρκετή υπομονή και πολλές επεμβάσεις στα άκρα της πλακέτας, καταφέραμε να την στριμώξουμε στο πλαστικό κουτί του ανεμιστήρα. Εντάξει, λυγίσαμε μερικά εξαρτήματα παραπάνω από όσο θα έπρεπε, αλλά τουλάχιστον τα καταφέραμε!



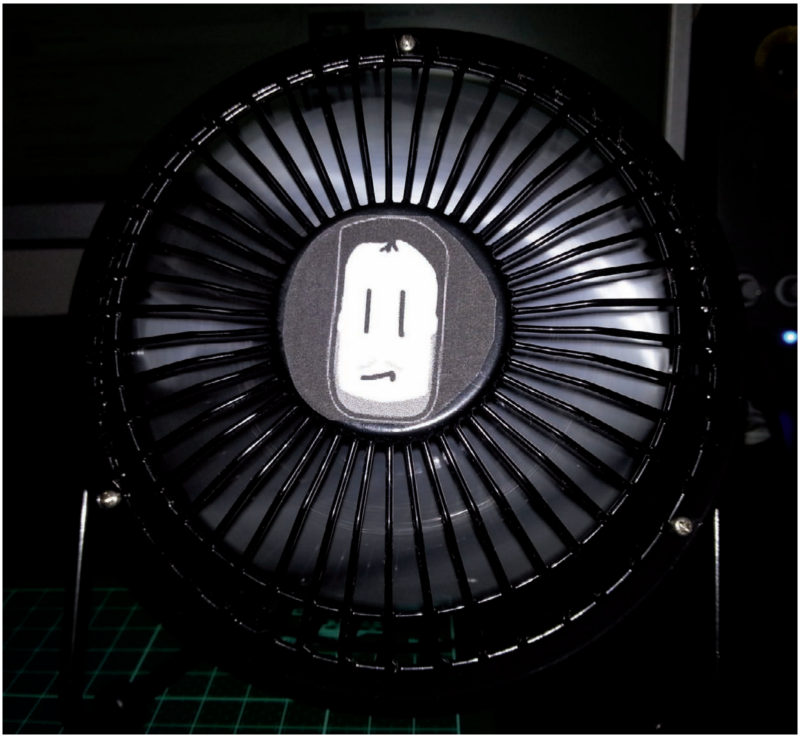
Αφού ανεβάσουμε το πρόγραμμα στον ATtiny, μπορούμε να το συνδέσουμε σε μια ελεύθερη θύρα USB. Μετά από λίγες στιγμές όμως τα Windows θα εμφανίσουν μήνυμα ότι η συσκευή δεν εγκαταστάθηκε με επιτυχία, γιατί δεν υπάρχουν drivers! Ευτυχώς, αυτό το πρόβλημα λύνεται πολύ εύκολα. Στον κατάλογο της βιβλιοθήκης libusb υπάρχει ένα εργαλείο που κατασκευάζει αυτόματα τους οδηγούς για οποιαδήποτε συσκευή USB, ώστε τα προγράμματα με τη βιβλιοθήκη libusb να επικοινωνούν μαζί της χωρίς προβλήματα. Αναφερόμαστε στο inf-wizard.exe. Αφού το τρέξουμε, επιλέγουμε τη δική μας συσκευή (ξεχωρίζει από

το DEVICE_ID με όνομα USB-fan) και στο επόμενο βήμα ορίζουμε το Manufacturer name (στην περίπτωσή μας δίνουμε το giannoug). Τέλος, πατάμε Next κι αποδεχόμαστε την εγκατάσταση του νέου driver. Αυτό ήταν όλο!

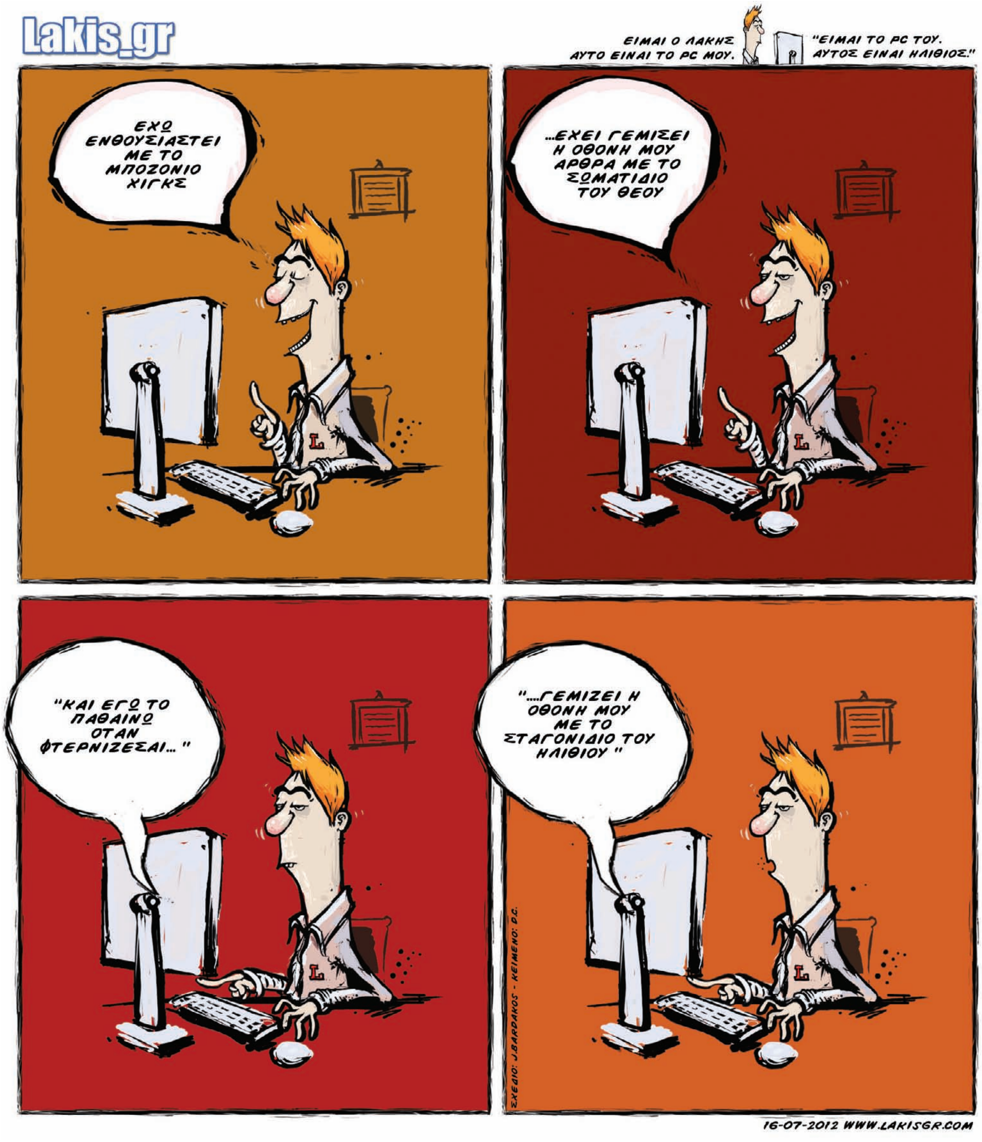
Wrap it up

Τα καταφέραμε! Με τα πολλά, ο ανεμιστήρας μας έβαλε μυαλό. Μάλιστα τα Windows έχουν φορτώσει τον σχετικό driver, ενώ και το προγραμμάτακι για τον έλεγχο του ανεμιστήρα από τον υπολογιστή είναι επίσης έτοιμο. Το μόνο που απομένει, είναι ο ...καλλωπισμός της συσκευής!

Για το σκοπό αυτό θα πρέπει να μεταφέρουμε το κύκλωμά μας από το breadboard (όπως το είδαμε στη σελίδα 97 του τεύχους 010) σε μια πλακέτα. Εδώ, οι επιλογές είναι πολλές. Ωστόσο ο πιο εύκολος κι ανέξοδος τρόπος είναι να χρησιμοποιήσουμε ένα stripboard. Πρόκειται για πλακέτες που διαθέτουν ένα μεγάλο grid από τρυπούλες, για τη τοποθέτηση των εξαρτημάτων. Αυτές οι τρυπούλες συνδέονται πάνω στην πλακέτα με γραμμές χαλκού. Αν όλα αυτά σας φαίνονται ακατανόητα και τ' ακούτε για πρώτη φορά, μπορείτε και να τα ξεχάσετε. Κανείς δεν θα σας κατηγορήσει. Ωστόσο, αν έχετε την οικειότητα ή τη θέληση να πειραματιστείτε, θα σας προτείναμε να αγοράσετε μερικές stripboards κι ένα κολλητήρι και ν' αρχίσετε τις δοκιμές! Στην αρχή ενδέχεται να μην μεταφέρετε σωστά το κύκλωμα, αλλά είμαστε σίγουροι ότι με λίγη υπομονή θα τα καταφέρετε. Εξάλλου, το κίνητρο για να πετύχετε τη μεταφορά του κυκλώματος σε πλακέτα, είναι πολύ ισχυρό: Θα αποκτήσετε έναν όμορφο και έξυπνο ανεμιστήρα, ο οποίος θα αναβοσβήνει από τον υπολογιστή!



Η τελευταία πινελιά ήταν να μπει το λογότυπό μας :D


Lakis_gr
ΕΙΜΑΙ Ο ΛΑΚΗΣ
ΑΥΤΟ ΕΙΝΑΙ ΤΟ PC ΜΟΥ.
"ΕΙΜΑΙ ΤΟ PC ΤΟΥ.
ΑΥΤΟΣ ΕΙΝΑΙ ΗΛΙΘΙΟΣ."
ΕΧΩ ΕΝΘΟΥΣΙΑΣΤΕΙ ΜΕ ΤΟ ΜΠΟΖΟΝΙΟ ΧΙΓΚΣ
...ΕΧΕΙ ΓΕΜΙΣΕΙ Η ΟΘΟΝΗ ΜΟΥ ΑΡΘΡΑ ΜΕ ΤΟ ΣΩΜΑΤΙΔΙΟ ΤΟΥ ΘΕΟΥ
"ΚΑΙ ΕΓΩ ΤΟ ΠΑΘΑΙΝΩ ΟΤΑΝ ΦΤΕΡΝΙΖΕΣΑΙ..."
"...ΓΕΜΙΖΕΙ Η ΟΘΟΝΗ ΜΟΥ ΜΕ ΤΟ ΣΤΑΓΟΝΙΔΙΟ ΤΟΥ ΗΛΙΘΙΟΥ"
ΣΧΕΔΙΟ: J.BARDAKOS - ΚΕΙΜΕΝΟ: D.C.
16-07-2012 WWW.LAKISGR.COM

deltahacker.gr